



UNIVERSIDAD MAYOR  
FACULTAD DE CIENCIAS

# “Diseño e Implementación de un Sistema de Monitoreo y Control para la elaboración de Cerveza Artesanal”

Proyecto de titulación para optar al título de Ingeniero Civil Electrónico

Yamil Abdala Maluje Villanueva

Santiago de Chile  
Diciembre de 2016



UNIVERSIDAD MAYOR  
FACULTAD DE CIENCIAS

## “Diseño e Implementación de un Sistema de Monitoreo y Control para la elaboración de Cerveza Artesanal”

Proyecto de titulación para optar al título de Ingeniero Civil Electrónico

Alumno: Yamil Abdala Maluje Villanueva

Profesor Guía: Francisco Rivera

Santiago de Chile

Diciembre de 2016

## Agradecimiento

Por todo el apoyo y fuerza:

Familia

“Catadores”

<IEEE>

Tesla almighty

SOLO USO ACADÉMICO

# INDICE

<b>RESUMEN .....</b>	<b>IV</b>
<b>ABSTRACT.....</b>	<b>VI</b>
<b>CAPITULO I.....</b>	<b>1</b>
<b>INTRODUCCION .....</b>	<b>1</b>
1.1. ANTECEDENTES .....	1
1.2. DESCRIPCION DEL PROYECTO.....	1
1.3. OBJETIVOS .....	3
1.3.1. Objetivo General .....	3
1.3.2. Objetivo Especifico.....	3
<b>CAPITULO II.....</b>	<b>4</b>
<b>ESTADO DEL ARTE .....</b>	<b>4</b>
2.1. ELABORACION CERVEZA ARTESANAL.....	4
2.1.1. Proceso.....	6
2.1.2. Problemas a abordar.....	9
2.1.3. Variables fundamentales.....	11
2.2. ANALISIS DE LA SITUACION ACTUAL DEL MERCADO .....	12
2.2.1. Introducción.....	12
2.2.2. Brewie .....	12
2.2.3. Brew-Boss .....	13
2.2.4. Grainfather .....	14
<b>CAPITULO III.....</b>	<b>16</b>
<b>DISEÑO DEL SISTEMA.....</b>	<b>16</b>
3.1. INTRODUCCION .....	16
3.2. DIAGRAMA DE BLOQUES SIMPLIFICADO .....	17
3.3. DIAGRAMA DE FLUJO SIMPLIFICADO DEL CONTROLADOR .....	18
3.4. DIAGRAMA DE BLOQUES DETALLADO .....	21
3.5. DIAGRAMA DE FLUJO DETALLADO DEL CONTROLADOR .....	23
3.5.1. Introducción .....	23
3.5.1.1. Parámetros .....	23
3.5.1.2. Calentando .....	25
3.5.1.3. Macerado.....	27
3.5.1.4. Lavado.....	29
3.5.1.5. Hervido .....	31
3.5.1.6. Enfriado .....	33
3.5.1.7. Traspaso Fermentador .....	35
3.5.1.8. Fermentador .....	37
3.5.1.9. Proceso .....	39

3.5.1.10. Temp Estable .....	41
3.5.1.11. PID .....	43
3.5.1.12. Datos .....	44
3.5.1.13. Alerta .....	46
<b>CAPITULO IV.....</b>	<b>48</b>
<b>IMPLEMENTACION DEL SISTEMA.....</b>	<b>48</b>
4.1. INTEGRACION DEL SISTEMA .....	48
4.1.1. Introducción .....	48
4.1.2. Estructura del Sistema .....	48
4.1.3. Controlador .....	50
4.1.4. Sensores.....	51
4.1.4.1. Temperatura.....	51
4.1.4.2. Peso .....	52
4.1.4.3. Alcohol.....	53
4.1.4.4. Flujo.....	54
4.1.4.5. Nivel .....	55
4.1.5. Actuadores.....	56
4.1.5.1. Válvula Solenoide.....	56
4.1.5.2. Bombas .....	57
4.1.5.3. Calentadores.....	58
4.1.5.4. Alimentador de Lúpulos.....	59
4.1.6. HMI .....	61
4.2. PRUEBAS .....	63
4.2.1. Introducción .....	63
4.2.2. Desarrollo.....	63
<b>CAPITULO V.....</b>	<b>71</b>
<b>ANALISIS DE INVERSIONES Y COSTOS.....</b>	<b>71</b>
<b>CAPITULO VI.....</b>	<b>74</b>
<b>CONCLUSIONES .....</b>	<b>74</b>
6.1. ANALISIS DEL DISEÑO IMPLEMENTADO .....	74
6.2. DIFICULTADES ENCONTRADAS .....	74
6.3. TIEMPO DEL DESARROLLO DEL PROTOTIPO .....	74
6.4. RESULTADOS Y TRABAJOS FUTUROS .....	75
<b>BIBLIOGRAFIA .....</b>	<b>77</b>
<b>ANEXOS .....</b>	<b>78</b>
ANEXO N°1. CÓDIGO DE LAS DIFERENTES FUNCIONES.....	78
ANEXO N°2. ....	104
ILUSTRACIONES DEL HMI EN DIFERENTES ETAPAS DEL PROCESO. ....	104
ANEXO N°3. ....	112

MONTAJE DE LOS DIFERENTES ACTUADORES Y SENSORES. ....	112
ANEXO N°4. ....	116
HOJA DE DATOS DE LOS SENSORES Y ACTUADORES DEL SISTEMA. ....	116
ANEXO N°5. ....	129
DATOS DIVERSOS QUE RESPALDAN ALGUNOS CÁLCULOS REALIZADOS.....	129
<b>INDICE DE TABLAS.....</b>	<b>130</b>
<b>INDICE DE FIGURAS .....</b>	<b>131</b>

SOLO USO ACADÉMICO

## RESUMEN

Este Proyecto de Título tiene como objetivo presentar el desarrollo de un prototipo de un sistema de monitoreo y control de las variables principales que influyen en la elaboración de cerveza artesanal. El propósito es facilitar su fabricación, acercar la elaboración de cerveza artesanal a las personas y servir como puente al momento de transformar una producción casera a una profesional (PYME). Este desarrollo involucra el diseño del hardware, software y posteriormente la implementación de un prototipo.

Para esto, se consideró el monitoreo de la cantidad de etanol liberado en la fermentación, el control de la temperatura, tiempo de cada proceso, cantidad de agua y peso del líquido a lo largo de diversas etapas.

Para lograr todo esto, se utilizó como Controlador del sistema, el microcontrolador ATmega2560 (Arduino Mega) el cual contendrá el código fuente necesario para monitorear y controlar todos los procesos a lo largo de la elaboración de cerveza artesanal. De igual manera, se implementó una pantalla táctil como interfaz humano-máquina para su fácil interacción.

Se presentaron diversos inconvenientes a lo largo de este proyecto, siendo la plomería el área más problemática. Debido a esto se debió reiterar numerosas veces el diseño de éste.

El sistema permite controlar y/o monitorear las variables fundamentales con gran precisión a lo largo de todo el proceso, almacenando todos los datos para su posterior análisis.

Se logró autonomía parcial del proceso, pudiendo desligar lo máximo posible al usuario final. También se logró diseñar e implementar sistemas de seguridad en diferentes etapas de la elaboración.

Finalmente, se diseñó e implementó una interfaz gráfica, amigable y fácil de utilizar, la cual muestra los parámetros e ilustraciones necesarias en todo el proceso de elaboración.

SOLO USO ACADÉMICO

## ABSTRACT

This project aims to present the development of a system prototype for monitoring and control of key variables that influence the development of craft beer, in order to facilitate manufacture, bringing the development of craft beer to people and serve as a bridge when transforming a home production to a professional one (PYME). This development involves the design of hardware and software and finally the implementation of a prototype.

For this, monitoring the amount of ethanol released in the fermentation and control of: temperature, time of each process, water quantity and weight of liquid along various stages was considered.

To achieve this, the microcontroller ATmega2560 (Arduino Mega) was used and with the source code uploaded is possible to monitor and control all processes along the production of craft beer. Also, a touch screen was implemented as human-machine interface for easy interaction.

There were several drawbacks throughout this project, with plumbing being the most problematic area. Due to this, it was necessary to reiterate numerous times the design of this one.

The system allows controlling and/or monitoring the fundamental variables with great precision throughout the entire process, storing all the data for later analysis.

Partial autonomy of the process was achieved, being able to untie as much as possible the end user. It was also possible to design and implement security systems in different stages of development.

Finally, it was possible to design and implement a graphical interface, friendly and easy to use, which shows the necessary parameters and illustrations throughout the entire process.

# CAPITULO I

## INTRODUCCION

### 1.1. ANTECEDENTES

La cerveza es una de las bebidas fermentadas más antigua de la humanidad. Se cree que ya existía en Sumeria (Mesopotamia) ya que se encontró una tablilla con grabados, la cual hacía menciones de la cerveza. Pero la cerveza no se limitó solo a Mesopotamia, los egipcios también desarrollaron la cerveza de una forma similar. A partir de una masa de pan sin hornear que dejaban fermentar en agua y gracias a las temperaturas y levaduras nativas, se convertía en cerveza. Al igual que en otras partes del mundo, existían otras bebidas parecidas a la cerveza que se fermentaban a partir de otros tipos de granos. En el imperio inca hacían la "chicha" a partir del maíz, los chinos hacían un tipo de cerveza llamado "Kiu" en base a trigo, cebada, mijo y arroz, e incluso en la antigua Britania elaboraban cerveza a base de trigo malteado antes de que los romanos introdujeran la cebada.

Se puede apreciar que la cerveza tiene diversas variaciones de ingredientes y métodos de elaboración pero se compone principalmente de cuatro ingredientes: agua, cereales (generalmente cebada o trigo), levadura y lúpulo (agregados recientemente). La combinación de cantidad y especie de cada uno, generan grandes variedades y estilos.

### 1.2. DESCRIPCION DEL PROYECTO

A medida que ha pasado el tiempo, se ha refinado cada vez más los ingredientes y el proceso de elaboración de esta bebida fermentada, logrando una inmensidad de tipos de cerveza, los cuales poseen características únicas, de color, sabor, amargor, aroma, entre otras. En la actualidad, existe un incremento en el consumo y en la elaboración de esta bebida alcohólica. Este último conlleva a un problema enorme, en cuanto al conocimiento y complejidad de elaboración para la persona que quiera indagar en este arte. Éste, requiere de un vasto conocimiento del proceso, al igual que una dedicación de su tiempo, constante monitoreo y control de las variables fundamentales a lo largo de todo su proceso, causando posibles errores humanos, lo cual perjudica la calidad del producto final.

Las grandes empresas productoras han mitigado este problema implementando sistemas complejos, los cuales se encargan del proceso de forma autónoma. Este método llega a ser inasequible (costoso) o complejo de utilizar, afectando a los pequeños productores al no poder implementarlas en su sistema.

Existen sistemas que son capaces de monitorear y controlar estas variables a lo largo del proceso. Pero estos poseen dos grandes problemas. Primero, se está obligado a elaborar una cierta cantidad de cerveza por vez, lo cual limita la capacidad de expansión a futuro. Segundo, se requiere de una constante supervisión por parte del cervecero, obligado a estar pendiente en ciertas etapas del proceso para realizar ajustes u otra actividad.

Fue esta razón la cual despertó el interés en investigar y desarrollar un prototipo de un sistema que monitorea y controla las variables fundamentales que influyen en este proceso, disminuyendo los errores humanos, logrando así, una elaboración de cerveza sencilla y precisa, al alcance de cualquiera.

Éste proyecto de título, evaluará un diseño capaz de monitorear y controlar las principales variables que influyen directamente en la producción de cerveza artesanal por medio de la implementación de sensores y actuadores, conectados a un dispositivo de control central, el cual se encargará de realizar los cálculos y acciones pertinentes para la correcta elaboración de la cerveza artesanal.

### **1.3. OBJETIVOS**

#### **1.3.1. Objetivo General**

Diseño e implementación de un prototipo de un sistema de monitoreo y control para variables fundamentales en la producción de cerveza artesanal.

#### **1.3.2. Objetivo Especifico**

- Recopilación de antecedentes que influyen en el proceso de elaboración de la cerveza artesanal. Análisis de los sistemas de producción actuales, estudiando sus características y modo de funcionamiento.
- Evaluación de alternativas de diseño de hardware y software para el sistema de monitoreo y control.
- Especificación de sensores, actuadores y micro-controladores, observando las alternativas disponibles comercialmente, teniendo presente la seguridad del sistema como criterio de selección.
- Diseñar una interfaz hombre máquina, intuitiva y fácil de usar, que sea capaz de controlar las variables planteadas de forma autónoma y manual.
- Diseño, implementación y pruebas del sistema de monitoreo y control propuesto, generando un prototipo de laboratorio funcional.
- Análisis de costos del prototipo.

## CAPITULO II

### ESTADO DEL ARTE

#### 2.1. ELABORACION CERVEZA ARTESANAL

Como se mencionó anteriormente, la cerveza posee cuatro ingredientes esenciales: malta, lúpulo, levadura y agua. Mientras que estos ingredientes son comunes, la variación de estos, más las diferentes técnicas de elaboración, producen una alta gama de estilos de cervezas, cada una con características únicas. Es por esto que la elaboración de cerveza artesanal, se considera un arte el cual cada maestro cervecero tiene sus técnicas y secretos para generar lo que él considera una obra maestra.

A continuación, se dará una pequeña introducción de los diferentes ingredientes, sus diferentes tipos, las características que otorgan, entre otros aspectos.

#### **Malta**

El malteado, es el proceso el cual el cereal de cebada es germinado controladamente y luego es secado rápidamente mediante air caliente, obteniendo la *malta*. Este proceso es complejo, se controlan y monitorean diversas variables, pero para el fin de este trabajo este proceso se explicará de forma sencilla.

Primero, se saturan los granos con agua, con el fin de “engañar” al grano a comenzar el crecimiento. Esto provoca que el grano comience a producir enzimas, las cuales convierten el almidón en azúcares con el fin de que se ocupe como energía para crear raíces y tallos. Es aquí donde se debe detener el proceso de germinado. Éste se logra secando rápidamente el grano con aire caliente, tostándolo. Esto preserva las enzimas y almidones almacenados en el grano, como al igual logra introducir sabores y colores adicionales al grano. Finalmente, la malta es limpiada de raíces, para su posterior uso. Este proceso no solo se limita a la cebada sino también se realiza a otros tipos de granos como trigo, avena, centeno, entre otros (Ironsides, 2015).

Debido a las variaciones de malteado, existe una gran variedad de maltas, las cuales cada una posee características únicas que afectan el color, sabor y dulzor de la cerveza.

## Agua

Más del noventa por ciento de la cerveza es agua, es por esta razón que una buena cerveza comienza con una excelente agua. Modificar las características del agua requiere conocimientos de química y cálculos lo cual, si se logra hacer bien, una cerveza artesanal promedio se puede transformar en una excelente cerveza. El agua se encuentra integrada por sales, las cuales, los aniones<sup>1</sup> que causan mayor impacto en el sabor de la cerveza, son los sulfatos, cloruro y sodio.

Las grandes cantidades de sulfatos presentes en el agua aumentan el amargor del lúpulo, frescura y sequedad, lo cual acentúa las características del lúpulo. Al igual que las grandes cantidades de cloruro o sodio en el agua, acentúan el carácter y dulzor de la malta (Ironsides, 2015).

## Lúpulo

El lúpulo es una planta, oriunda de Europa, Asia occidental y Norteamérica. En la elaboración de la cerveza, se utiliza la flor hembra, la cual otorga amargor, sabores complejos y aromas distintivos. Inicialmente, se utilizaban las flores secas, pero éstas son muy sensibles a la oxidación, lo cual solo se podían almacenar por periodos cortos de tiempo. En la actualidad se elaboran extractos líquidos o polvos comprimidos, logrando así un mejor rendimiento y estabilidad del producto.

El lúpulo posee diversos componentes de los cuales, destacan solo tres:  $\alpha$ -ácidos<sup>2</sup>, otorga el amargor en la cerveza.  $\beta$ -ácidos<sup>3</sup>, otorga el aroma en la cerveza. *Aceites esenciales*, otorga sabor y aroma. Por lo tanto, existen diversos tipos de lúpulos los cuales poseen variaciones de éstos, los cuales, a su vez, resaltan, aroma, sabor y amargor, dependiendo del uso dado (Ironsides, 2015).

---

<sup>1</sup> (Atkins, Jones, & Gismondi, 2006)

<sup>2</sup> (Ragoni, 2016)

<sup>3</sup> (Ragoni, 2016)

## Levadura

La levadura es un organismo clasificado como hongos microscópicos unicelulares y su principal función en la elaboración de la cerveza es la fermentación del mosto. Existen una inmensidad de especies de levadura, pero solo dos son responsables de hacer cerveza: *Levadura Ale* y *Levadura Lager*.

La primera, es la especie que se utiliza para fermentar la cerveza de tipo Ale. Esta, es la misma levadura que se utiliza para hacer pan. Este tipo de levadura prefiere temperaturas tibias, normalmente 20°C (12°C y 21°C) y varían según sus características y desempeño. Además de lograr fermentar el mosto, algunas se utilizan para refinar y limpiar la cerveza (en su fermentación) como otras las usan para generar sabores frutales y más intensos.

En cambio, el segundo tipo de levadura, fermenta a temperaturas más frías (8°C y 14°C), y la mayoría se utiliza para refinar y otorgar un carácter más suave a la cerveza (Ironsides, 2015).

### 2.1.1. Proceso

A pesar de que se considere un arte, éste sigue una estructura básica de elaboración, la cual se compone de cuatro pasos esenciales: maceración, cocción, enfriado y fermentación. Cada paso posee variaciones en ingredientes, procesos y ejecución, entre otras.

La maceración, consiste en poner en contacto el agua con la malta molida, permitiendo que las enzimas degraden los constituyentes de la malta (carbohidratos y proteínas) a formas solubles, generando el líquido fermentable, denominado *mosto*. El mosto se debe calentar a una temperatura específica, lo cual depende de aspectos como, el cereal utilizado, las características y el tipo de cerveza a elaborar.

En la maceración ocurren diversas reacciones químicas, las cuales se producen en diferentes condiciones de pH, tiempo y temperatura. Por lo tanto, la variación de esos parámetros afecta en las características de la cerveza artesanal (Hernández, 2003).

Se puede realizar el proceso de maceración principalmente de tres formas diferentes.

La primera es con *extracto de malta*<sup>4</sup>. Este método es una forma de comenzar a hacer cerveza artesanal de forma sencilla, ya que elimina el proceso de generar los tipos de azúcares necesarios en el mosto.

El segundo método, utiliza un híbrido, el cual usa extracto de malta con otras maltas especiales, otorga cuerpo, color, dulzor y sabor al mosto.

El tercer método, es el más completo y complejo de todos ya que se comienza desde cero. Éste utiliza malta en granos molidos, los cuales junto con el agua, a una determinada temperatura, libera el almidón no fermentable de la malta en azúcares fermentables. La temperatura del agua varía dependiendo del tipo de azúcar fermentable deseado. A este método se le debe agregar un paso extra como lo es la *clarificación* o *filtración* del mosto. Éste consiste en separar el mosto del grano con el fin de generar un mosto libre de residuos. Para lograr esto, se debe seguir generalmente los siguientes sub-pasos: *mashout*, *recirculación* y *lavado*. *Mashout*, es la acción de elevar la temperatura del mosto por sobre la de maceración (aproximadamente 77°C), con el fin de detener la conversión enzimática del almidón, logrando un mosto más fluido. La *recirculación*, consiste en sacar mosto desde el fondo del recipiente y agregarlo delicadamente en la parte superior, generando un filtro natural en la superficie del recipiente (ya que se encuentra los granos de malta flotando en su superficie). Finalmente, el *lavado* consiste en agregar agua a una temperatura específica a los granos destilados previamente (ya que se debe levantar los granos del mosto, dejándolos en su superficie hasta que no gotee más) con el fin de extraer la mayor cantidad de azúcares fermentables del grano como al igual lograr un reajuste en la densidad del mosto.

Con los azúcares extraídos y completamente disueltos se procede a la siguiente etapa, la cocción. Aquí se procede a llevar el mosto a la ebullición, lo cual logra diversas

---

<sup>4</sup>(Ironsides, 2015)

reacciones bioquímicas, como: isomeración<sup>5</sup> del lúpulo, la desinfección<sup>6</sup> del mosto y la precipitación de la proteína. También, se procede a agregar los lúpulos, los cuales, dependiendo del tiempo de adición al mosto, la cantidad agregada y el tipo de lúpulo, otorgan amargor, sabor y/o aroma a la cerveza.

Luego de terminada la cocción, se procede a enfriar el mosto lo más rápido posible hasta la temperatura de fermentación. Esto se debe a dos razones fundamentales. La primera, es desinfectar el mosto, ya que, entre 20 y 40°C proliferan, con particular intensidad, los gérmenes indeseables presentes en la cerveza, causantes de sabores y aromas no deseados por lo que debe estar el menor tiempo posible expuesto a estos gérmenes. La segunda razón, es debido a que el mosto posee componentes que más tarde pueden manifestarse produciendo turbidez, lo cual, el enfriamiento rápido, precipita tales componentes, facilitando su posterior extracción (Ironsides, 2015).

Finalizado el enfriado, se procede a la etapa de fermentación. Es aquí donde el azúcar presente en el mosto, principalmente se convierte en alcohol y dióxido de carbono por medio de la levadura. Este proceso ocurre en diversas etapas. Cabe destacar que el crecimiento de la levadura y el proceso de fermentación, implica un complejo proceso metabólico que se escapa de los alcances de este proyecto. Debido a esto, este proceso se explicará de forma general y concisa.

Este proceso comienza al momento que se agrega la levadura al mosto. Una vez realizado esto, comienza la siguiente etapa. En ésta, la levadura se prepara para su reproducción. Esto se logra mediante el uso de oxígeno y nutrientes presentes en el mosto, lo cual dura aproximadamente entre un par de horas o en algunos casos un día. Esto depende de la cantidad de levadura añadida y temperatura del mosto. A una mayor temperatura y cantidad de levadura, resulta un tiempo más corto de esta etapa (considerando mantenerse dentro del rango permitido por la levadura). Pasada las veinticuatro horas se debe observar el burbujeo del airlock<sup>7</sup> y la formación de una capa

---

<sup>5</sup> (Mathews, Van Holde, Ahern, & González de Buitrago, 2002)

<sup>6</sup> (Higieneyan, 2016)

<sup>7</sup> DISPOSITIVO EL CUAL PERMITE LA SALIDA DEL DIÓXIDO DE CARBONO DURANTE EL PROCESO DE FERMENTACIÓN, PERO IMPIDE EL INGRESO DE AIRE AL INTERIOR DEL RECIPIENTE, ASEGURANDO UNA FERMENTACIÓN ESTÉRIL.

de espuma (mezcla de levadura y proteínas del mosto), la cual posee un color cremoso y pequeñas manchas verde-café. Si estos dos síntomas no se presentan dentro de las primeras cuarenta y ocho horas, la cantidad de levadura fue insuficiente y es necesario agregar más. Una vez formada la capa de espuma, la levadura se encuentra creciendo exponencialmente y comienza consumiendo los azúcares más simples como la glucosa o fructosa y luego pasa a los azúcares más complejos como la maltosa. El tiempo de esta etapa depende de diversas variables de fermentación, siendo la temperatura la variable clave. Finalmente, comienza la última etapa en la cual la levadura hace un proceso de limpieza, reabsorbiendo sub productos de la fermentación, los cuales normalmente no se desean en la cerveza. En cuanto a lo visual, se aprecia la desaparición gradual de la capa de espuma, como también la decantación de las células de levadura al fondo del recipiente fermentador. El tiempo de esta etapa puede variar de tres a diez días (en algunos casos más), dependiendo de las condiciones de fermentación. Para confirmar el término de esta última etapa, se debe tomar lecturas consecutivas de la densidad del líquido con el fin de asegurarse que ha alcanzado el objetivo planteado inicialmente en la receta (Ironsides, 2015).

Se puede apreciar que la elaboración de cerveza posee diversos métodos y técnicas, los cuales cada maestro cervecero opta por escoger, manteniendo constante las cuatro etapas principales: maceración, cocción, enfriado, fermentado y variando los sub procesos dentro de cada etapa principal, ajustándolos a sus necesidades.

### **2.1.2. Problemas a abordar**

Como se puede observar, la elaboración de cerveza artesanal es un proceso extenso y complejo, posee diversos factores y métodos. Debido a esto, existe un sinnúmero de problemas lo cual afecta el producto final. Este proyecto de título se enfocó en los más importantes, descritos a continuación.

El principal problema que enfrenta el maestro cervecero, al momento de elaborar su cerveza artesanal, es el control de la temperatura. Este requiere tener un constante monitoreo y control a lo largo de todo el proceso, demandando su tiempo completo. Esto conlleva a que el más mínimo descuido provoque errores en el producto final.

El segundo problema al cual se enfrenta, es el tiempo. Es crucial esta variable, ya que influye directamente en el producto final. Ésta se debe controlar en cada una de las cuatro etapas de elaboración. Suele ocurrir que, debido al descuido por parte del maestro cervecero, la etapa en la cual se encuentra, se detiene en un tiempo mucho mayor de lo programado, afectando el producto final. Debido a esto, se debe controlar la cantidad de tiempo el cual se está en cada etapa.

El tercer problema al cual se enfrenta, es la ineficiencia o en algunos casos escasez en la toma de datos a lo largo de todo el proceso. Al momento de elaborar la cerveza no se toman datos de las variables fundamentales, lo cual conlleva a dos problemas principales. El primero, no se puede controlar de forma precisa cada una de ellas, por lo tanto, no se mantiene la temperatura estable, o no se respeta los tiempos correctos en el proceso. Segundo, no hay datos para su posterior análisis, provocando no poder obtener alguna mejora o encontrar un posible error en la receta.

Otro problema es el presente en la etapa de fermentación. Como se pudo apreciar, esta etapa es compleja, lo cual requiere de un conocimiento extenso y preciso, al igual que posee diversos factores que afectan su proceso, siendo el primordial, la temperatura. Ésta, se debe controlar de forma precisa con el fin de ayudar a la levadura con su trabajo de fermentación. Normalmente, en esta etapa, se deja el estanque fermentador en una habitación oscura, sin humedad y sin control de su temperatura. Lo que conlleva a una variación de su temperatura a lo largo del proceso y una insuficiencia de datos, los cuales confabulan al momento de tomar decisiones a futuro.

El objetivo de este proyecto de título es el de mitigar, de la mejor forma posible, los problemas mencionados anteriormente, logrando disminuir el error humano presente en la elaboración de la cerveza artesanal con el fin de obtener un producto de calidad, como también, poder ajustar la receta de forma precisa debido a que cuenta con la información pertinente a lo largo de todo su proceso.

### **2.1.3. Variables fundamentales**

En la elaboración de cerveza artesanal se enfrenta con una diversidad de factores y variables, los cuales afectan el producto final. Debido a esto, se analizó cuáles de ellos son fundamentales con el fin de controlarlos y/o monitorearlos.

Algunas de las variables que se mencionan a continuación solo se monitorean, ya que resulta muy difícil o de alto costo, controlarlas.

Como se ha ido mencionando a lo largo de este proyecto, la principal variable es la temperatura. Ésta, juega un rol importante a lo largo de todo el proceso de elaboración desde la maceración hasta la fermentación de la cerveza. Debe mantenerse estable para asegurar la consistencia de esta variable a todo momento. Por último, esta variable es dinámica, lo que implica que cambia su valor dependiendo de la etapa en la cual se encuentre.

Otra variable es el tiempo. Se debe controlar la cantidad de tiempo el cual se está en cada etapa con el fin de respetar la receta la cual se está elaborando.

La cantidad de agua también juega un rol importante en la elaboración de cerveza. Ésta debe mantenerse dentro de ciertos rangos establecidos por la receta elegida/creada con el fin de lograr la gravedad específica a la cual se desea llegar. Es por esto que se controla la cantidad de agua en dos etapas específicamente. La primera es en el inicio de la maceración. Se traspasa una cantidad predeterminada desde uno de los recipientes a otro con el fin de comenzar la maceración con la cantidad de agua específica. La segunda es en el término del macerado, específicamente en el lavado. Se traspasa otra cantidad calculada desde el primer al segundo recipiente con el fin de compensar la pérdida de agua que se produjo con los granos de malta y la futura pérdida con la ebullición del agua en la etapa de cocción.

El peso es otra variable fundamental. Se debe conocer la cantidad de agua evaporada en la etapa de cocción y la cantidad de agua que se pierde por la absorción de la malta al final del macerado. Con el fin de poder retribuirla en el lavado de la malta para mantener la relación de agua/malta constante.

Por último, está el etanol. Como se mencionó con anterioridad, en la etapa de fermentación, la levadura libera principalmente alcohol (en forma de etanol) y dióxido de carbono. Debido a que el sensor de alcohol es de bajo coste en comparación con el de dióxido de carbono, se decidió por monitorear el etanol. A lo largo de la etapa de fermentación, se va generando cada vez más etanol hasta que llegue a un máximo. Es aquí donde la levadura ya no posee mayores herramientas para convertir los azúcares presentes. Una vez ocurrido esto, el valor de etanol es relativamente constante. Por lo tanto, se puede determinar el fin de la etapa de fermentación.

## **2.2. ANALISIS DE LA SITUACION ACTUAL DEL MERCADO**

### **2.2.1. Introducción**

En la actualidad se encuentran dispositivos los cuales poseen características y formas diferentes. Pero todos ellos con un objetivo en particular, elaborar cerveza artesanal.

### **2.2.2. Brewie**

Brewie es un sistema de elaboración de cerveza artesanal autónomo (figura 1), creado por Marcell Pal y su equipo. Fue fundada gracias a la plataforma de financiación colectiva llamada “Indiegogo”, la cual recaudó más de \$720 mil USD en febrero del 2015.

El objetivo principal de este sistema es poseer una máquina compacta, (53.8cm alto, 32.5cm de fondo y 45cm de largo) doméstica, la cual sea fácil de utilizar, sea estéticamente atractiva y que tenga el fin de facilitar la elaboración de cerveza a las personas.

Este sistema se encarga de las tres primeras etapas de la elaboración, las cuales el sistema las realiza en forma totalmente autónoma, dejando la última (fermentación) por parte del usuario. Este sistema también cuenta con auto limpieza, aliviando el trabajo para el usuario (Indiegogo, 2015).

Figura 1 Brewie



Fuente: Denia, C. (2015). "Brewie, la máquina para hacer cerveza en casa de forma automática". Recuperado de <http://blogs.20minutos.es/la-gulateca/2015/02/19/brewie-la-maquina-para-hacer-cerveza-en-casa-de-forma-automatica/>

### 2.2.3. Brew-Boss

*Brew-Boss* (figura 2) es un sistema de elaboración de cerveza compacto, modular y de fácil armado. Existen versiones de 40, 57 y 76 litros de capacidad. Posee un filtro que ellos denominan **COFI** (Center Out Forced Infusion, por sus siglas en inglés), el cual, durante el macerado, mantiene todos los granos de malta en su interior pero dejando el agua fluir uniformemente. También posee un controlador, que se encarga de controlar la temperatura del agua y el tiempo en sus diferentes etapas.

El objetivo principal de este sistema es su alta capacidad de adaptarse. Este puede ser un complemento al sistema que uno posee, agregando solo las piezas necesarias con el fin de poder elaborar cerveza artesanal (Boss, 2016).

Figura 2 Brew-Boss



Fuente: Brew-Boss (2016). Recuperado de <http://www.brew-boss.com/>

#### 2.2.4. Grainfather

*Grainfather* (figura 3) es un sistema todo en uno, compacto con capacidad de treinta litros, bomba de recirculación y doble elemento calentador. Posee una caja de control que se encarga del control de la temperatura en cada etapa del proceso. Éste también posee un intercambiador a contraflujo para ser utilizado en la etapa de enfriado (Grainfather, 2016).

Figura 3 Grainfather



Fuente: Grainfather. Recuperado de <http://www.grainfather.com/shop/grainfather/grainfather.html>

SOLO USO ACADÉMICO

## CAPITULO III

### DISEÑO DEL SISTEMA

#### 3.1. INTRODUCCION

A lo largo del tiempo se han creado diversos diseños de sistemas para elaborar cervezas artesanales, los cuales cada uno de ellos cuenta con diversas características. Estos diseños se diferencian entre ellos en aspectos como cantidad de recipientes, tamaño del sistema, formas de fabricación, facilidad de limpieza, cantidad de partes y accesorios, costo de materiales, facilidad de uso, capacidad de producción, entre otras cosas. Debido a esto, se deben tomar ciertas consideraciones al momento de diseñar el sistema tanto en hardware como en software.

En el hardware, se debió considerar ciertos criterios al momento de diseñar el sistema. La complejidad de fabricación es un criterio importante. Debido a que se cuenta con habilidades básicas de mecánica y plomería, se requiere de un diseño fácil de fabricar. Se debe utilizar materiales comunes de encontrar en el mercado nacional e internacional, considerando la facilidad de poder reemplazarlos al momento de alguna falla. Además, se optó por utilizar una pantalla táctil con el fin de interactuar con el usuario de forma fácil e intuitiva, entregando toda la información de forma clara y concisa.

En el software, la mayor variación (con lo descrito en el capítulo anterior) se presenta en la etapa de macerado. Es aquí donde se escogieron ciertos sub procesos. Se seleccionó el método más complejo mencionado en el capítulo anterior, el cual involucra extraer los diferentes azúcares de los granos de malta molidos. Este método necesitaba de ciertos subprocesos de clarificación, lo que conlleva a realizar el “*mashout*”, recirculación y lavado. El recirculado se realiza a lo largo de todo el proceso de macerado, mediante la activación de la bomba de recirculación y el solenoide de recirculado, logrando clarificar lo más posible el mosto a lo largo de todo el proceso. En cuanto al lavado, se realiza como lo mencionado anteriormente. Finalmente, el *mashout*, se optó por no realizarlo directamente. Esto quiere decir, que no se mantiene a la temperatura de *mashout* por un tiempo predeterminado sino que, luego del lavado,

se pasa directamente al hervido, el cual eleva su temperatura inmediatamente, realizando un *mashout*, indirectamente.

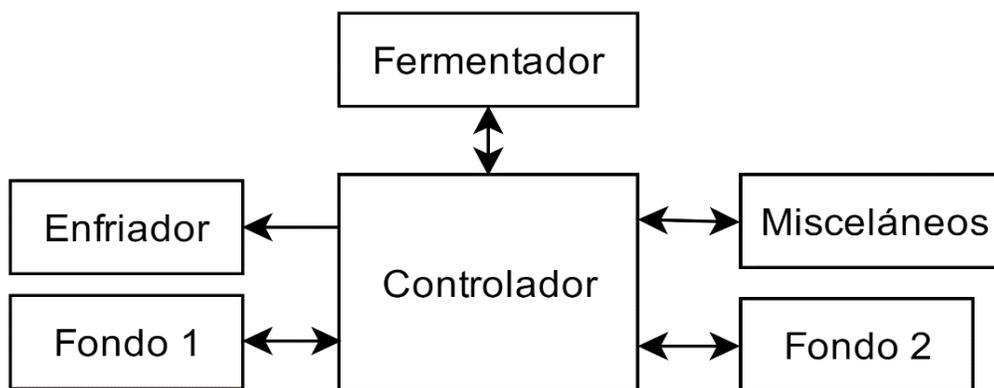
También, se agregaron unas etapas a lo largo del proceso con el fin de facilitar la elaboración. Una de estas etapas es el calentamiento. Éste consta de calentar previamente el agua del primer recipiente a la temperatura de macerado con el fin de verterlo en el segundo recipiente, en donde se encuentra la malta, para comenzar el proceso de macerado con la temperatura correcta. La segunda etapa agregada es el denominado “Traspaso Fermentador”. Éste simplemente vacía el mosto ya enfriado al recipiente de fermentado, midiendo la cantidad de mosto que se obtuvo. Por último, se agregó la etapa denominada “Alerta”. Ésta se encarga de revisar el funcionamiento de los sensores de flujo, como también los sensores de nivel, con el fin de arrojar una alerta si existe un mal funcionamiento de éstos.

### **3.2. DIAGRAMA DE BLOQUES SIMPLIFICADO**

A continuación, se muestra el diagrama en bloques simplificado del sistema (figura 4). Este sistema consta principalmente de seis bloques: *Fondo 1*, *Fondo 2*, *Controlador*, *Fermentador*, *Enfriador* y *Misceláneos*. Cada bloque realiza una función específica con el fin de concretar cada etapa del proceso de elaboración de cerveza artesanal.

Este sistema se compone de un controlador centralizado el cual se encarga de monitorear y controlar los elementos periféricos que lo componen, presentando diferentes señales de comunicación. Algunas señales son bidireccionales ya que se necesita conocer los estados de los sensores con el fin de activar los respectivos actuadores. En cambio, otras señales son unidireccionales ya que se componen solo de actuadores como lo es el bloque de *enfriador*.

Figura 4 Diagrama en bloques simplificado del sistema



Fuente: Propia (2016)

### 3.3. DIAGRAMA DE FLUJO SIMPLIFICADO DEL CONTROLADOR

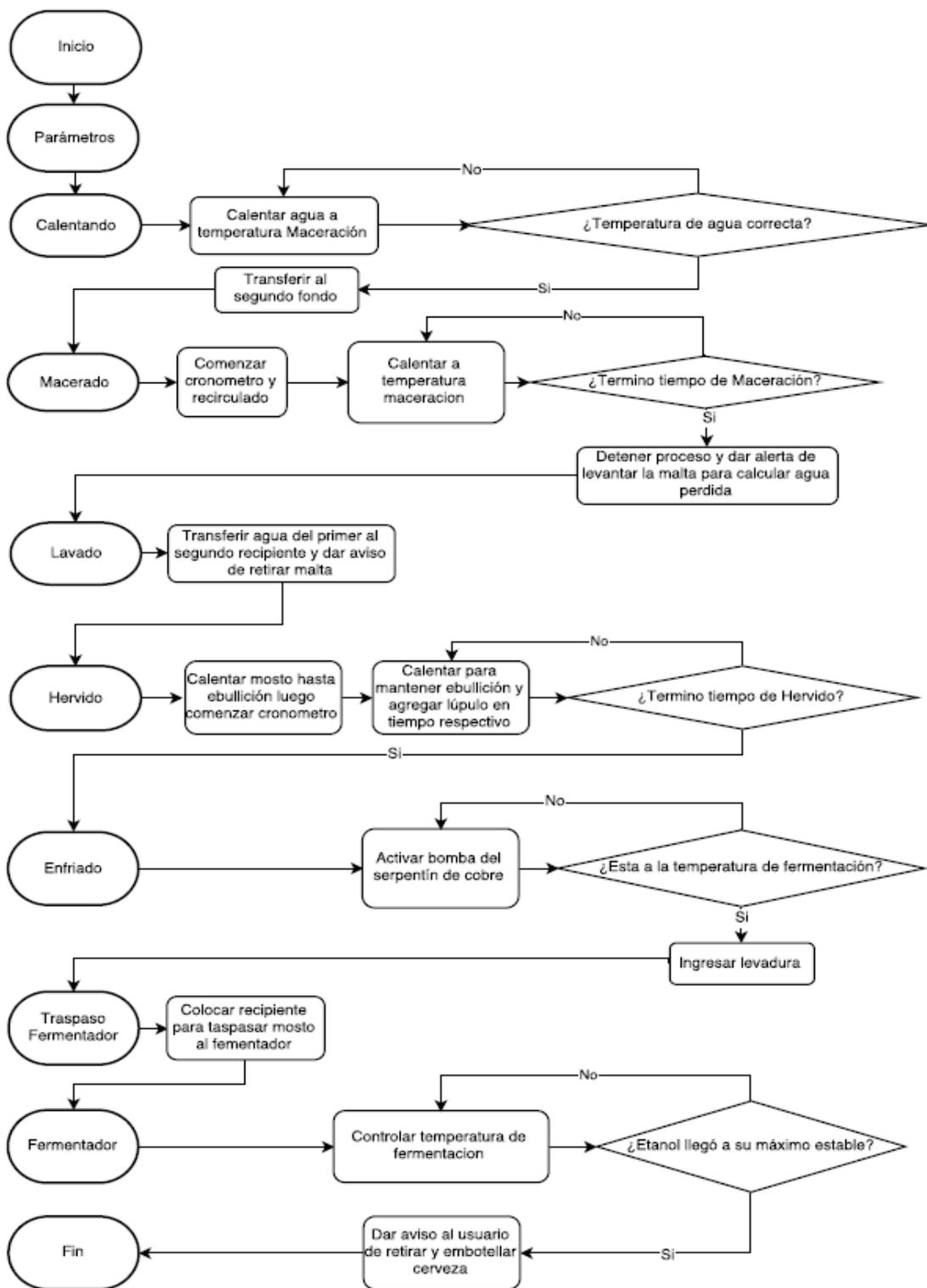
A continuación, se muestra el diagrama simplificado del controlador (figura 5). Se puede apreciar que el proceso es secuencial, el cual abarca todas las etapas de la elaboración de la cerveza artesanal. Éste posee procesos extras, los cuales son necesario para el correcto funcionamiento. Se explicará de forma breve cada etapa del proceso, debido a que más adelante se detallará con mayor profundidad.

El proceso comienza con el ingreso de los parámetros de la cerveza que se desea elaborar. Una vez ingresados, el sistema comienza por calentar el agua ubicada en el primer recipiente a la temperatura de maceración. Logrado esto, el sistema comienza a transferir la cantidad necesaria de agua al recipiente dos en el cual se encuentra la malta. Luego de estabilizada la temperatura, comienza la etapa de macerado. Esto implica que se activa el cronómetro como también el recirculado del mosto. Luego de terminado el tiempo de macerado, el sistema se detiene y da aviso de levantar la malta para poder medir y calcular el peso perdido por la absorción de la malta. Terminado esto, comienza el lavado, donde se transfiere agua desde el recipiente uno al dos, extrayendo la mayor cantidad de azúcar posible de la malta. A continuación, da aviso al usuario de retirar la malta para dar inicio a la etapa de hervido o cocción. Aquí se eleva la temperatura del mosto hasta llegar a hervirlo. Logrado esto, se activa el cronómetro de esta etapa. A lo largo de esta etapa se deben ingresar los diferentes lúpulos mediante la activación del alimentador de lúpulos en el tiempo definido.

Terminado el hervido, comienza el proceso de enfriado, el cual consiste en hacer pasar agua fría por el serpentín de cobre ubicado en el recipiente dos con el fin de enfriar el mosto a la temperatura de fermentación. Una vez alcanzada la temperatura de fermentación, el sistema ingresa la levadura al mosto. Finalizado esto se debe traspasar el mosto al recipiente de fermentación, dando aviso al usuario de colocar el recipiente de fermentación. Finalmente comienza la etapa de fermentación que se realiza de forma independiente al sistema, controlando la temperatura del mosto y monitoreando la producción de etanol hasta que este último llegue a un máximo estable, dando la alerta de que la cerveza esta lista para ser embotellada.

SOLO USO ACADÉMICO

Figura 5 Diagrama de flujo simplificado del controlador

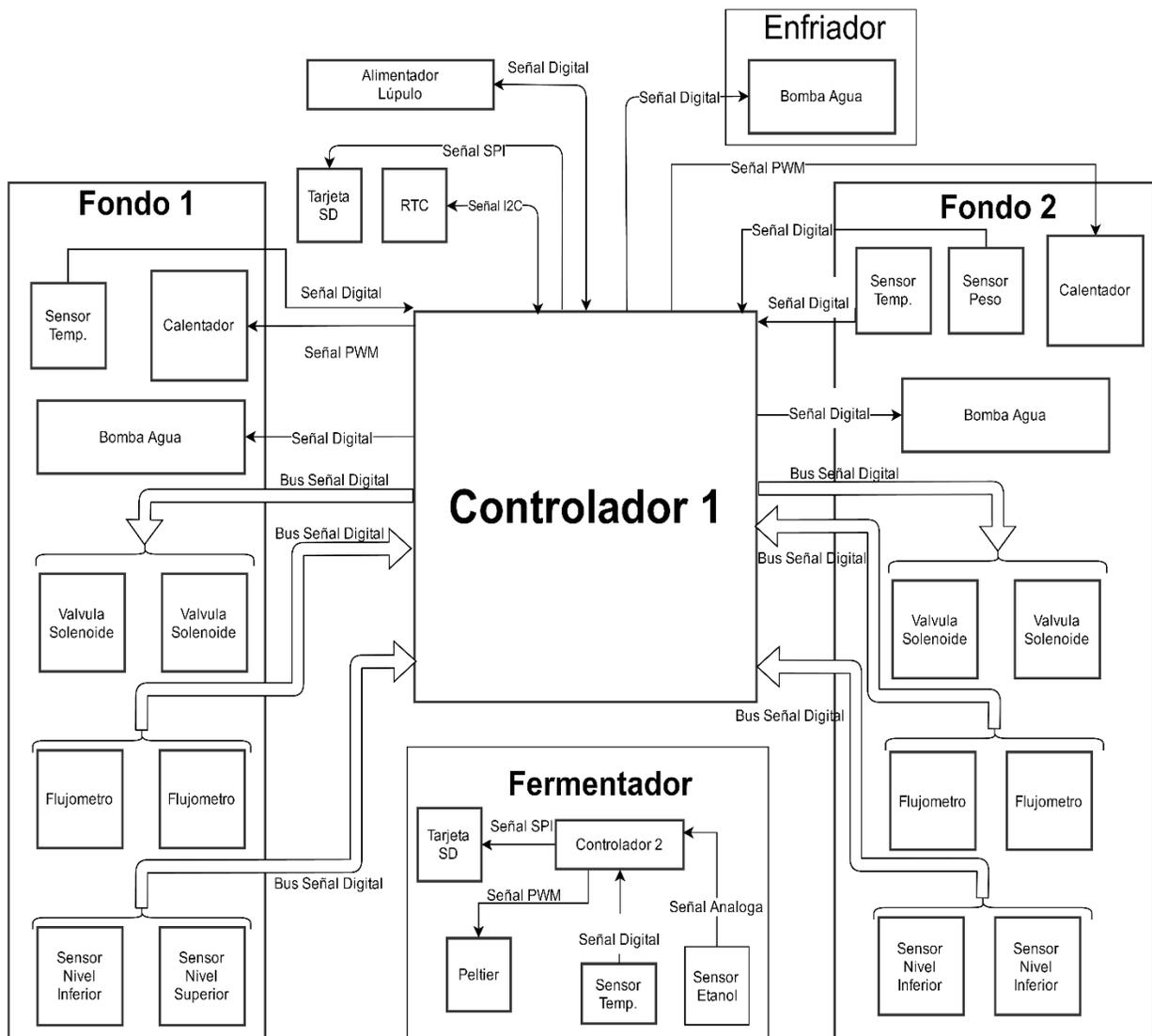


Fuente: Propia (2016)

### 3.4. DIAGRAMA DE BLOQUES DETALLADO

A continuación, se aprecia el diagrama de bloques del sistema (figura 6). Se puede apreciar que posee una gran variedad de sensores y actuadores, los cuales cada uno posee una función específica. Aquí se mencionarán los diferentes actuadores y sensores, pero en el siguiente capítulo se detallarán con profundidad cada uno de ellos.

Figura 6 Diagrama de Bloques



Fuente: Propia (2016)

Este sistema consta principalmente de cinco bloques. *Fondo 1*, *Fondo 2*, *Controlador*, *Fermentador* y *Enfriador*.

El *Fondo 1* posee exactamente los mismos actuadores y sensores que el *Fondo 2*, con la excepción del sensor de peso, el cual se encarga de medir el peso del recipiente dos. Este bloque posee un sensor de temperatura digital, encargado de tomar muestras al momento de recircular el líquido por sus mangueras; la bomba de recirculación (categoría alimentario); el elemento calentador que se encarga de calentar el líquido; las válvulas de solenoide, usadas para hacer pasar el líquido por diferentes circuitos de mangueras; los sensores de flujo, encargados de determinar si existe flujo en las mangueras y de medir la cantidad de líquido que fluye y por último, los sensores de nivel inferior y superior, encargados de entregar alarmas si el nivel de líquido es incorrecto.

El bloque *Controlador* se encarga de recibir y mandar las señales pertinentes con el fin de controlar y monitorear los diferentes dispositivos conectados a él, actuando como el “cerebro” de todo el sistema.

El *Enfriador* se compone de una bomba de agua el cual se encarga de realizar la recirculación de agua fría al serpentín de cobre.

El bloque *Fermentador* es un sistema independiente del resto. Posee su propio controlador el cual se encarga de las variables que afectan solamente al proceso de fermentación. Esto se realizó así debido a que brinda la opción de poder elaborar cerveza tantas veces como se desea mientras que paralelamente se esté fermentando las cervezas en el fermentador sin interrupción del proceso. Este bloque posee un sensor de temperatura digital encargado de tomar muestras de temperatura al fermentador con el fin de poder controlar posteriormente el *peltier* (Garcia, 2013) para enfriar o calentar el espacio. También, posee un sensor de alcohol encargado de tomar muestras de la cantidad de etanol que produce la fermentación de la cerveza con el fin de tener un criterio de finalización del proceso de fermentado. Por último, posee una

tarjeta SD (Secure Digital, por sus siglas en inglés) para guardar el registro de todas las variables a lo largo del proceso.

El alimentador de lúpulo es otro componente del sistema, encargado de arrojar los lúpulos al segundo recipiente.

Por último, se encuentra la tarjeta SD, encargada de registrar los datos de todas las variables fundamentales a lo largo del proceso y el módulo RTC, encargado de suministrar la hora respectiva con el fin de registrarlo en la tarjeta SD.

### **3.5. DIAGRAMA DE FLUJO DETALLADO DEL CONTROLADOR**

#### **3.5.1. Introducción**

A continuación, cada función del controlador se dividió en diversos diagramas de flujo detallados, logrando así una comprensión más sencilla del proceso.

##### **3.5.1.1. Parámetros**

Esta es la primera función<sup>8</sup> que se ejecuta (figura 7). Aquí se muestran por pantalla (Anexo N°2 (1)) los diferentes parámetros que debe ingresar el usuario, como lo son la temperatura y tiempo de maceración (Anexo N°2 (1A)), tiempo de cocción con sus respectivos tiempos de los lúpulos (Anexo N°2 (1B-C)), relación agua/malta (Anexo N°2 (1D)), entre otras cosas. El sistema cuenta la cantidad de veces que se presionó el botón “*next*”, con el fin de conocer cuándo se ingresaron todos los parámetros.

Luego si se cumple la condición anterior se procede a pesar la malta. El sistema reinicia el valor del sensor de peso, luego el usuario debe colocar la cantidad de malta deseada en el recipiente dos. Al mismo tiempo el sistema muestra en pantalla el peso de la malta ingresada. Luego el usuario debe presionar el botón “*next*” para confirmar el peso y continuar el proceso.

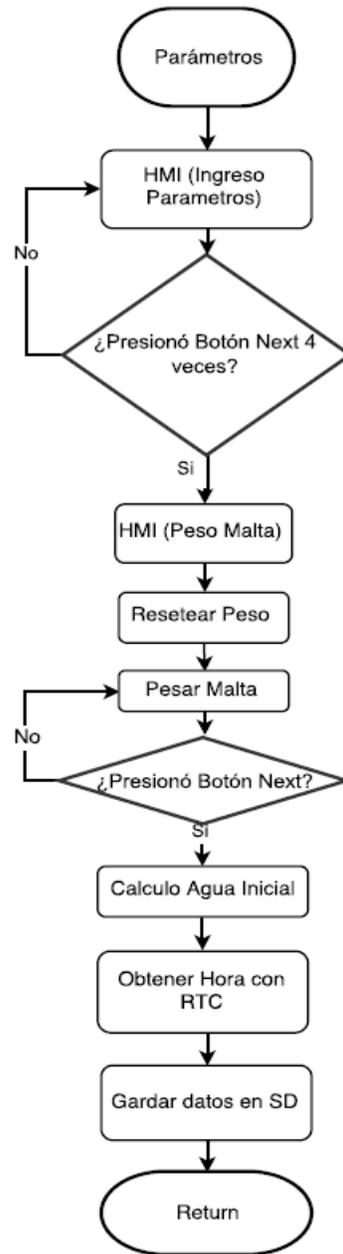
Presionado el botón, el sistema calcula la cantidad de agua necesitada en la maceración con las variables de relación y peso de la malta, luego obtiene la hora

---

<sup>8</sup> Anexo N°1 (1) código Parametros.

cuando comienza la elaboración de la cerveza, mediante el uso del RTC, para finalmente guardarlos en la tarjeta SD.

Figura 7 Función Parámetros



Fuente: Propia (2016)

### 3.5.1.2. Calentando

Luego de terminada la función anterior, se prosigue a la función “Calentando”<sup>9</sup> (figura 8). Aquí se muestra por pantalla la ilustración gráfica de diferentes procesos (Anexo N°2 (2)). Primero que está recirculando el agua. Segundo que se está calentando el agua del primer recipiente. Tercero, la temperatura actual del mismo recipiente. Finalmente, en la esquina superior izquierda se muestra si se grabó o no los datos en la tarjeta SD (este ícono estará presente a lo largo de todo el proceso de elaboración de cerveza artesanal). Posteriormente, se activan los actuadores y sensores pertinentes a esta etapa para luego ingresar a la siguiente función denominada “Proceso” (explicado en profundidad en el punto 3.5.9).

Aquí, se activa el calentador del primer recipiente, lo cual lleva la temperatura del agua a la temperatura de maceración. Una vez estable esa temperatura se da por finalizado la función.

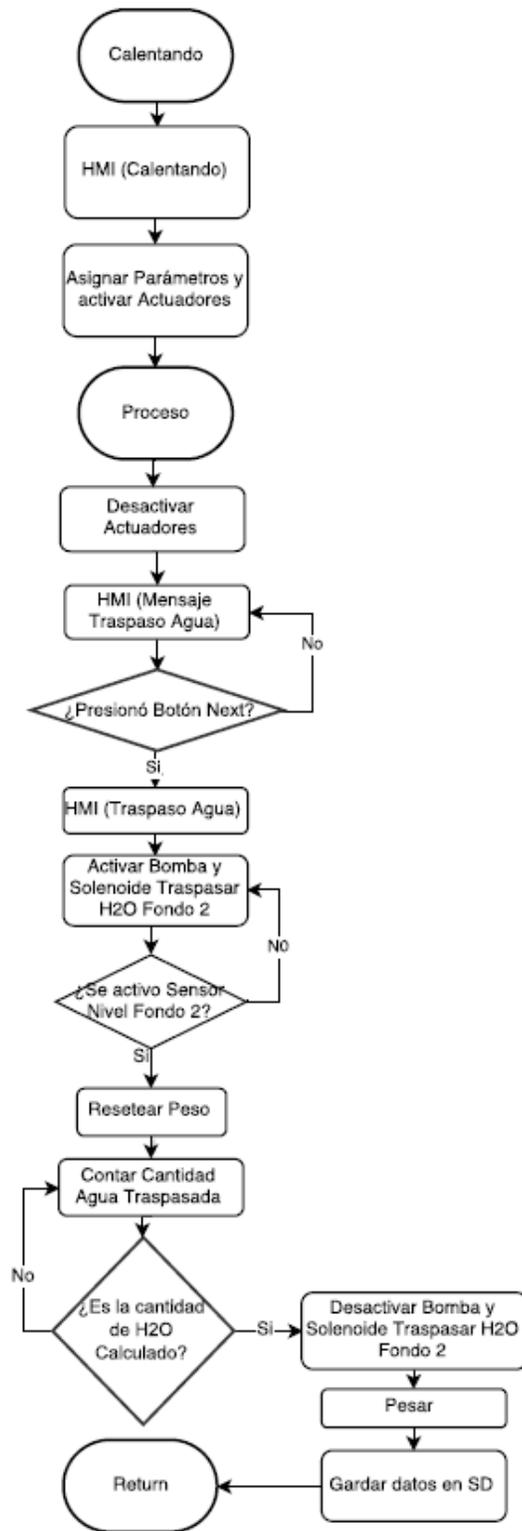
Una vez finalizado la función “Proceso”, se deben desactivar los actuadores y sensores pertinentes. Luego, se muestra en pantalla la cantidad de agua calculada para ser usada en la etapa de maceración (Anexo N°2 (3)). En seguida, el usuario debe presionar el botón “next” para dar inicio al traspaso de esa agua para lo cual se muestra en pantalla la cantidad traspasada en tiempo real (Anexo N°2 (4)).

Para calcular la cantidad de agua traspasada, se optó por comenzar a medir el flujo de agua al momento que se active el sensor de nivel inferior del segundo recipiente. Esto, como medida de seguridad para el calentador (siempre debe estar sumergido bajo líquido) y para mantener constante la relación malta/agua. Una vez que se activa el sensor de nivel inferior del segundo recipiente se resetea el sensor de peso y comienza a medir la cantidad de agua transferida. Este proceso se detiene al momento de que el sensor de flujo mide la cantidad de agua calculada. Luego el sistema pesa la cantidad de agua transferida, lo cual sería la cantidad de agua inicial del proceso de maceración. Finalmente, se guardan los datos en la tarjeta SD.

---

<sup>9</sup> Anexo N°1 (2) código calentando.

Figura 8 Función Calentando



Fuente: Propia (2016)

### 3.5.1.3. Macerado

Luego de terminada la función anterior, se procede a iniciar la función de “Macerado”<sup>10</sup> (figura 9). El sistema muestra en pantalla los dos recipientes con sus temperaturas respectivas (Anexo N°2 (5)). Estas temperaturas son representadas de tres colores distintos: azul, verde y rojo. Azul, significa que está por debajo de la temperatura a la cual se desea llevar. Verde, significa que está en la temperatura a la cual se desea llegar. Rojo, significa que está por sobre la temperatura a la cual se desea llegar. En la pantalla también se muestra la hora actual como también, el estado de la tarjeta SD (si se está grabando o no la información).

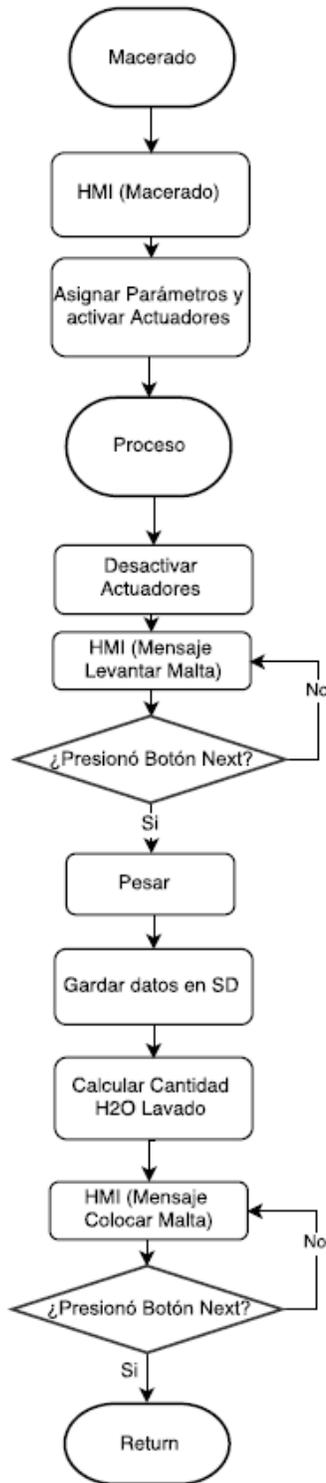
Posteriormente, se activan los actuadores y sensores pertinentes a esta etapa para luego ingresar a la función “Proceso”. Aquí, se activa el calentador del segundo recipiente, lo cual lleva la temperatura del agua a la temperatura de maceración. Una vez estable esa temperatura se da inicio al tiempo de maceración.

Terminado el tiempo de maceración, se deben desactivar los actuadores y sensores pertinentes. Luego, el sistema muestra un mensaje por pantalla de levantar la malta del recipiente (Anexo N°2 (6)). Al instante que el usuario levanta la malta y presiona el botón “next”, el sistema pesa el recipiente dos con el fin de estimar la cantidad de agua absorbida por la malta. Posteriormente, se guarda ese valor y se calcula la cantidad de agua que se debe utilizar en el proceso de lavado. Finalmente, se muestra en pantalla un mensaje de colocar nuevamente la malta para su posterior lavado (Anexo N°2 (7)).

---

<sup>10</sup> Anexo N°1 (3) código macerado.

Figura 9 Función Macerado



Fuente: Propia (2016)

#### **3.5.1.4. Lavado**

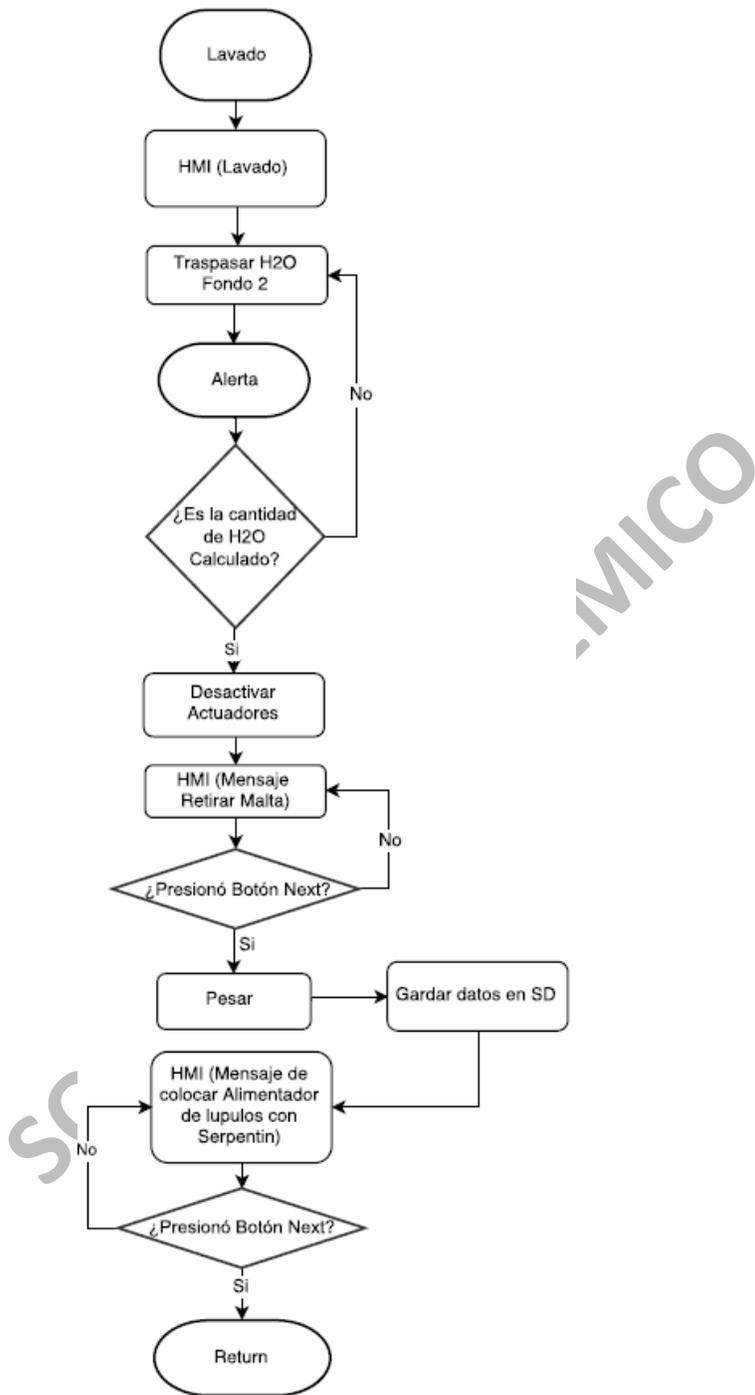
Una vez que el usuario presionó el botón “next”, se procede a comenzar la función de “Lavado”<sup>11</sup> (figura 10). El sistema muestra en pantalla el traspaso del agua en tiempo real para el proceso de lavado (Anexo N°2 (8)). Mientras se está traspasando agua del recipiente uno al dos, se está ingresando constantemente a la función de “Alerta” para dar mensajes de alerta si es que ocurre algún error en este proceso como también se está midiendo el agua traspasada mediante el sensor de flujo. Una vez traspasada la totalidad del agua, el sistema detiene los actuadores y muestra en pantalla un mensaje de retirar la malta (Anexo N°2 (9)).

Una vez presionado el botón “next”, se procede a pesar y guardar los datos en la tarjeta SD. Luego el sistema muestra en pantalla un aviso de colocar el serpentín y alimentador de lúpulos en su respectiva posición (Anexo N°2 (10)).

---

<sup>11</sup> Anexo N°1 (4) código lavado.

Figura 10 Función Lavado



Fuente: Propia (2016)

### **3.5.1.5. Hervido**

Luego que el usuario haya colocado el alimentador de lúpulos, el serpentín de cobre y haya presionado el botón “next” se da inicio a la función de “Hervido”<sup>12</sup> (figura 11).

Aquí se muestra en pantalla el segundo recipiente con su respectiva temperatura, como al igual el estado de cada lúpulo (si se ha ingresado o no), la hora actual, el recirculado del mosto y el tiempo faltante de la etapa. Luego, el sistema activa los actuadores y sensores pertinentes para luego ingresar a la función “Proceso”. A lo largo de esta función se activa el calentador para llegar a hervir el mosto. Cuando esto ocurre, comienza a contar el tiempo, lo cual muestra en pantalla el tiempo faltante para terminar el “Hervido”. Si es que es tiempo de agregar algún lúpulo el sistema activa el alimentador de lúpulos, botando el lúpulo al mosto. Además, aparece en pantalla la confirmación que se agregó el lúpulo con el símbolo de comprobación (Anexo N°2 (11)).

Una vez cumplido el tiempo de “Proceso”, se procede a desactivar los actuadores para dar término a la función de hervido.

---

<sup>12</sup> Anexo N°1 (5) código hervido.

Figura 11 Función Hervido



Fuente: Propia (2016)

### 3.5.1.6. Enfriado

Terminado el hervido, comienza la función “Enfriado”<sup>13</sup> (figura 12). Aquí se muestra por pantalla (Anexo N°2 (12)) el segundo recipiente con su temperatura actual, el recirculado y un copo de nieve (representando que se está enfriando el mosto).

Luego de mostrar en pantalla, se asignan los diferentes parámetros de esta función y se activan los diferentes actuadores y sensores.

Luego, se hace ingreso a la función “Alerta” con el fin de revisar si existe algún problema con algún sensor de flujo o sensor de nivel. Posteriormente, se debe adquirir la temperatura del sensor del segundo recipiente, como también la hora actual, con el fin de guardarlo en la tarjeta SD si es que han pasado diez segundos.

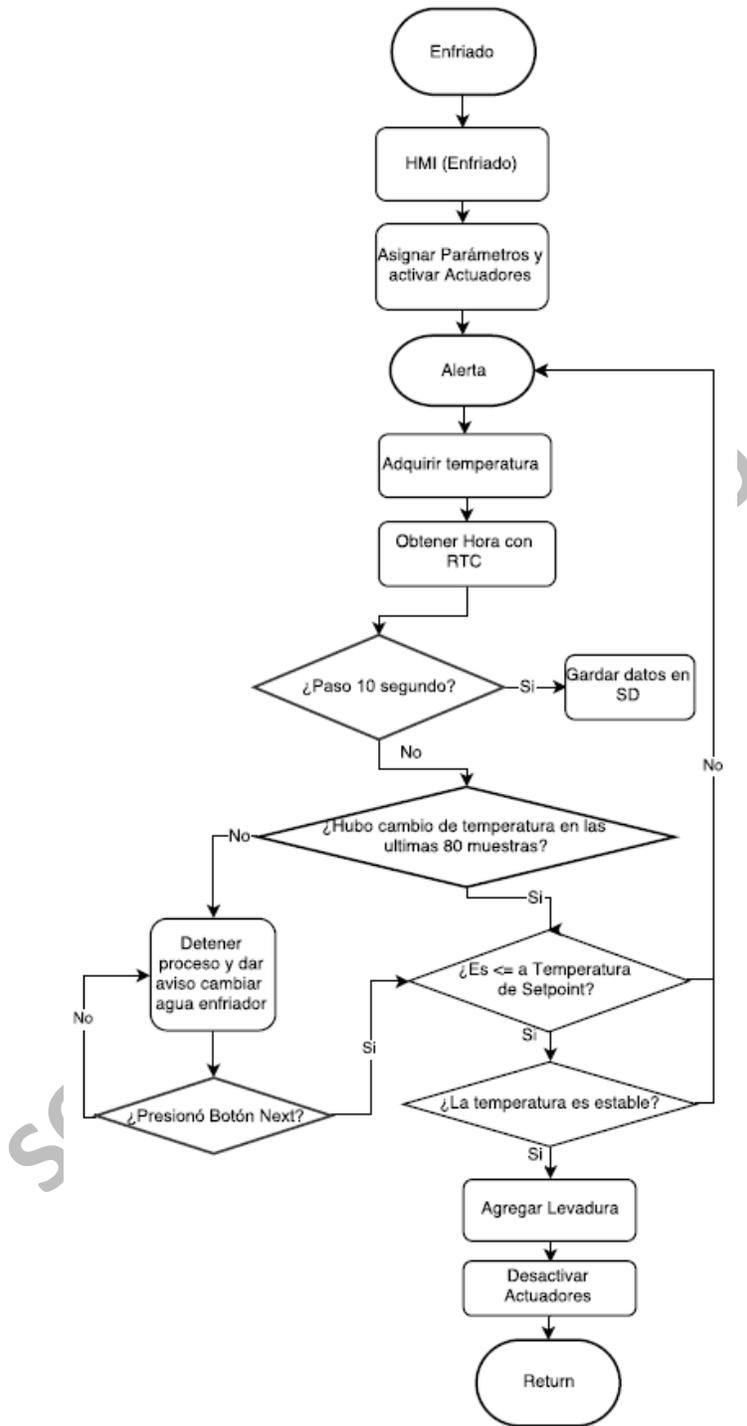
Luego se genera una función de alerta que verifica si hubo cambio de la temperatura del recipiente en un cierto periodo de tiempo. Esto se realizó debido a que, si no ha cambiado la temperatura dentro de las últimas ochenta muestras, se asume que el agua usada para enfriar el mosto ya no es capaz de enfriarlo velozmente (debido a que alcanzó o está cerca del equilibrio térmico (Morales, 2001)). Por lo tanto, el sistema entrega una alerta y detendrá el proceso, para que el usuario cambie el agua.

Luego, el sistema compara la temperatura actual del recipiente con la temperatura de fermentación. Si es menor y se mantiene estable, se debe agregar la levadura y desactivar los actuadores pertinentes para dar fin al proceso. Si es mayor, regresa a la función “Alerta” y repite el proceso descrito hasta que la temperatura actual sea menor a la de fermentación.

---

<sup>13</sup> Anexo N°1 (6) código enfriado.

Figura 12 Función Enfriado



Fuente: Propia (2016)

### 3.5.1.7. Traspaso Fermentador

Luego de enfriado el mosto, se debe traspasar al recipiente de fermentación. Esto se logra con la siguiente función denominada “Traspaso Fermentador”<sup>14</sup> (figura 13).

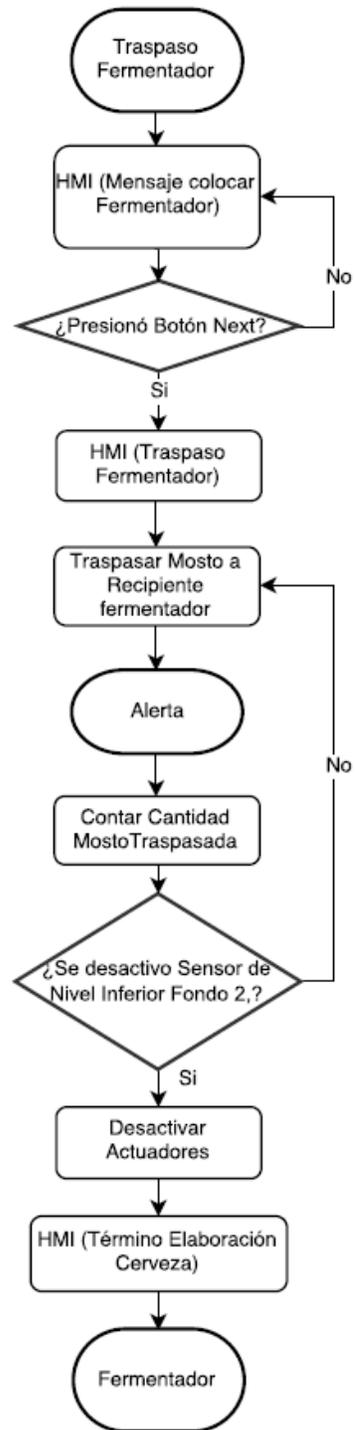
Aquí se muestra por pantalla un mensaje, el cual indica que el usuario debe colocar el recipiente fermentador (Anexo N°2 (13)). Una vez colocado el recipiente, el usuario debe presionar el botón “next”, lo que procede a mostrar por pantalla el traspaso del mosto. En pantalla se muestra el recipiente el cual contiene el mosto y el recipiente fermentador (representados por botellas), como también se muestra la cantidad (en tiempo real) de mosto que se está traspasando (Anexo N°2 (14)).

Para que ocurra esto, el sistema activa los actuadores y sensores pertinentes. Luego, ingresa a la función “Alerta” con el fin de analizar si existe alguna falla para detener el proceso y reportarlo. Después, el sistema debe contar la cantidad de mosto que se está transfiriendo mediante el uso del sensor de flujo. Este ciclo se repite hasta que se detecta que el sensor de nivel inferior, del recipiente dos, se desactiva, provocando que se detenga el proceso, desactivando los actuadores y sensores pertinentes.

---

<sup>14</sup> Anexo N°1 (7) código traspaso fermentador.

Figura 13 Función Traspaso Fermentador



Fuente: Propia (2016)

### **3.5.1.8. Fermentador**

Esta función (figura 14) se encuentra implementada en el segundo controlador del sistema ya que se optó por implementar la etapa de fermentación como un sistema independiente del resto del proceso con el fin de no restringir la elaboración de más cerveza artesanal.

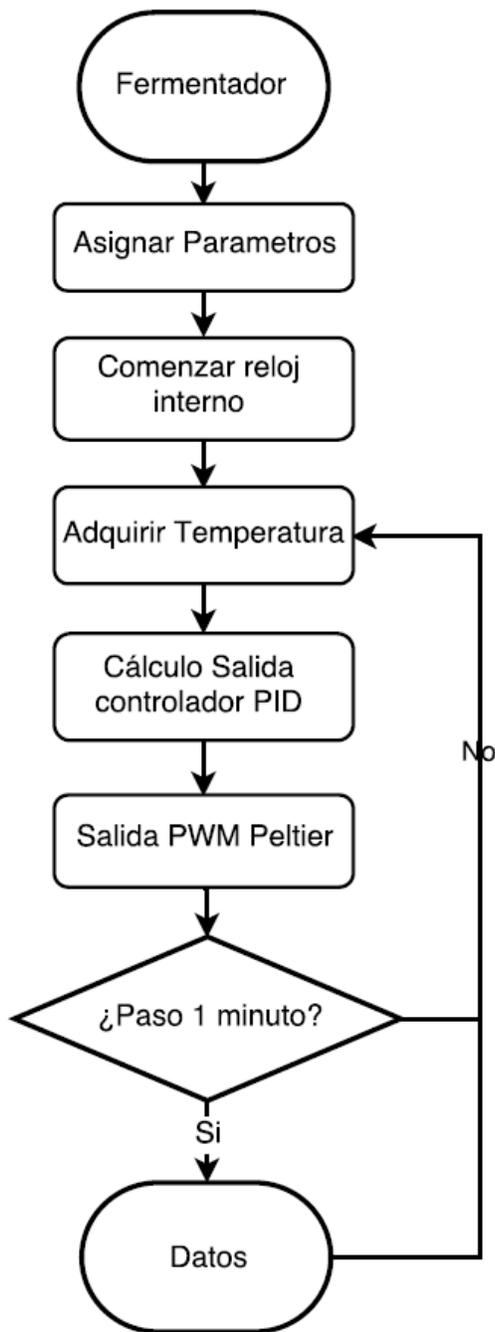
Esta función<sup>15</sup> comienza mediante la asignación de parámetros como lo es la temperatura del mosto, resolución del sensor, entre otras cosas. Luego, se da inicio al contador interno del controlador con el objetivo de contar cada minuto que transcurre. Posteriormente, el sistema adquiere la temperatura del sensor con el fin de poder calcular la salida del controlador PI para luego utilizar ese valor obtenido como salida PWM (Pulse Width Modulation, por sus siglas en inglés) en el dispositivo calentador/enfriador para ajustar la temperatura ambiente del fermentador.

Este ciclo se realiza permanentemente con el fin de mantener constante la temperatura de fermentación. Una vez transcurrido un minuto, el sistema ingresa a la función "Datos", el cual se encarga de obtener, revisar validez y guardar los datos, como al igual se encarga de dar aviso el término de la etapa de fermentación.

---

<sup>15</sup> Anexo N°1 (8) código fermentador.

Figura 14 Función Fermentador



Fuente: Propia (2016)

### 3.5.1.9. Proceso

Esta función<sup>16</sup> (figura 15) es utilizada en diversas etapas a lo largo del proceso de elaboración de cerveza.

Este proceso comienza llamando a la función "PID", la cual se encarga de alcanzar la temperatura deseada, utilizando un controlador PI (Mazzone, 2002). Luego, se debe obtener la hora actual, utilizando el RTC, para luego ingresar a la función de "Alerta". En seguida, el sistema evalúa si han pasado diez segundos, utilizando el reloj interno del controlador, para grabar los datos en la tarjeta SD.

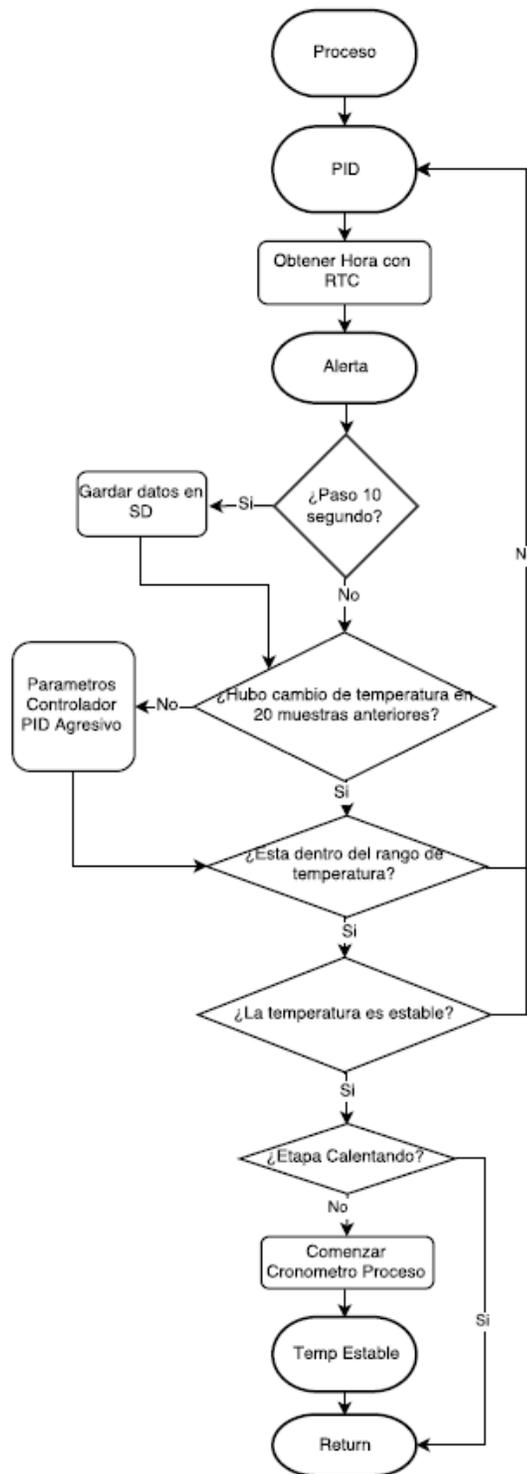
Más tarde se creó una condición la cual se encarga de ajustar los parámetros del controlador PI a un modo agresivo, aumentando sus valores. Esto se realizó debido a que, en diferentes pruebas realizadas, la temperatura no alcanzaba la esperada, debido a que el sistema no era capaz de contrarrestar las perturbaciones externas causadas por la temperatura ambiental. Esto ocurre debido a que, cuando se sintonizaron los parámetros del controlador PI, la temperatura ambiente variaba entre 18 [°C] y 23 [°C], aproximadamente. Por ende, cuando se realizaron las diferentes pruebas, la temperatura ambiente alcanzaba, aproximadamente los 11 [°C] y debido a que los recipientes son de aluminio que poseen un gran coeficiente de conductividad térmica (Netto, 2016), disipan una gran cantidad de energía en forma de calor al medio ambiente, ocasionando que los calentadores (con los parámetros de PI obtenidos) no logren contrarrestar esa perturbación.

Finalmente, si la temperatura es estable y no estamos en la etapa de "*Calentando*", se procede a comenzar el cronómetro del tiempo e ingresar al proceso de "*Temp Estable*".

---

<sup>16</sup> Anexo N°1 (9) código proceso.

Figura 15 Función Proceso



Fuente: Propia (2016)

### **3.5.1.10. Temp Estable**

Esta función<sup>17</sup> (figura 16) se encarga de las diferentes tareas que se requieren en la etapa de maceración o cocción, como lo es el tiempo, agregar los lúpulos, pesar el mosto, entre otras cosas.

Todo comienza ingresando a la función “*Alerta*” con el fin de ver posibles errores en el proceso.

Luego se ingresa a la función “*PID*” la cual se encarga de mantener la temperatura específica a lo largo de todo el proceso.

En seguida, se debe conocer en qué etapa se encuentra el proceso. Si es en el macerado, el sistema se encarga de mostrar todo lo necesario en la pantalla (explicado anteriormente). Si se encuentra en la cocción, el sistema debe mostrar todo lo necesario por pantalla (explicado anteriormente) y debe verificar si es tiempo de agregar algún lúpulo al mosto. Si es así, se activa el alimentador de lúpulos y se guarda la acción en la tarjeta SD.

Luego se debe pesar el mosto con el fin de mantener un registro del cambio de peso de éste y poder realizar un posterior análisis.

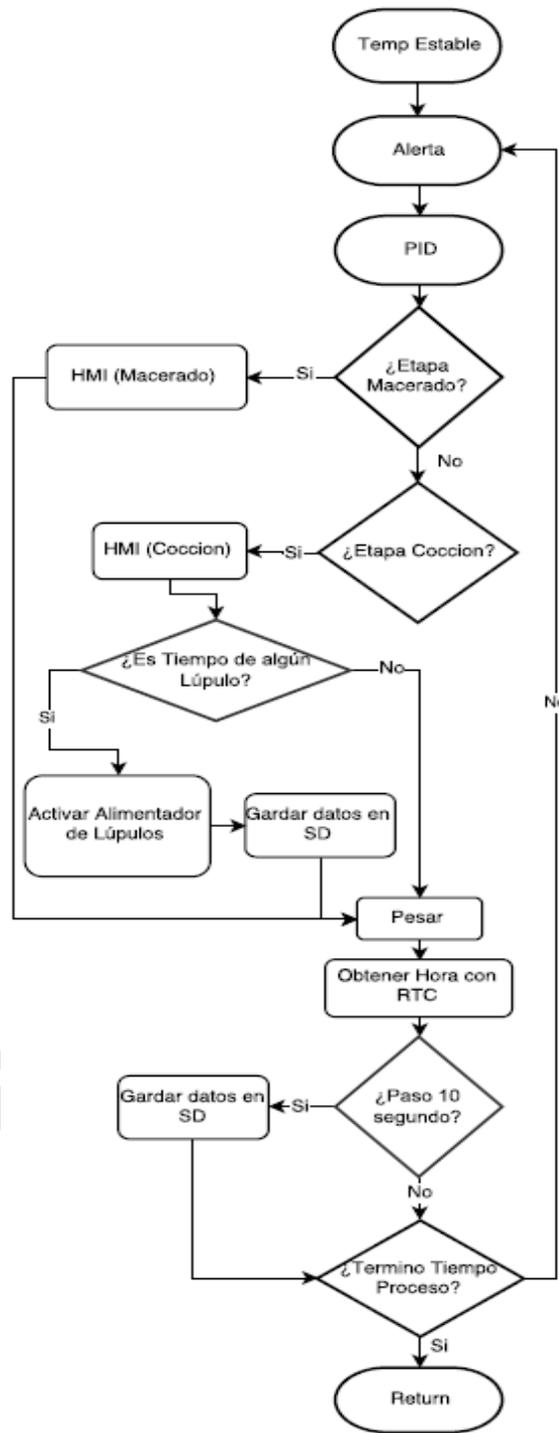
Posteriormente, se debe obtener la hora con el RTC y el sistema verifica si han pasado diez segundos de tal forma de poder grabar los datos en la tarjeta SD.

Finalmente, el sistema verifica si el tiempo del proceso termina en la etapa actual. Si no se ha finalizado, ingresa nuevamente a “*Alerta*”, repitiendo el ciclo nuevamente de lo contrario, finaliza esta función.

---

<sup>17</sup> Anexo N°1 (10) código temp estable.

Figura 16 Función Temp Estable



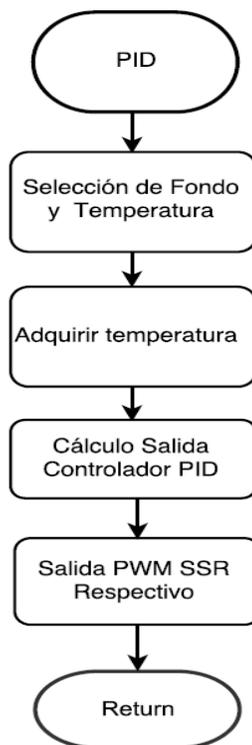
Fuente: Propia (2016)

### 3.5.1.11. PID

Esta función<sup>18</sup> (figura 17) se encarga de alcanzar y mantener la temperatura requerida en cada uno de los dos recipientes. Esto lo logra mediante el uso del controlador PI.

Primero, se debe conocer qué recipiente necesita un control de temperatura y a qué temperatura se desea mantener o alcanzar. Una vez establecido eso, se debe conocer la temperatura actual del recipiente escogido mediante el sensor de temperatura respectivo. Con los datos anteriores ya definidos, se realizan los cálculos pertinentes por medio de la librería PID (Beauregard, 2011), la cual calcula la salida del controlador. Finalmente, ese valor es utilizado para controlar el calentador de su respectivo recipiente por medio de una señal PWM.

Figura 17 Función PID



Fuente: Propia (2016)

<sup>18</sup> Anexo N°1 (11) código pid.

### 3.5.1.12. Datos

Esta función<sup>19</sup> (figura 18) se encuentra implementado en el controlador del fermentador. Ésta se encarga de guardar los datos obtenidos por el sensor de alcohol y de temperatura, como también, dar aviso del término del proceso de fermentación.

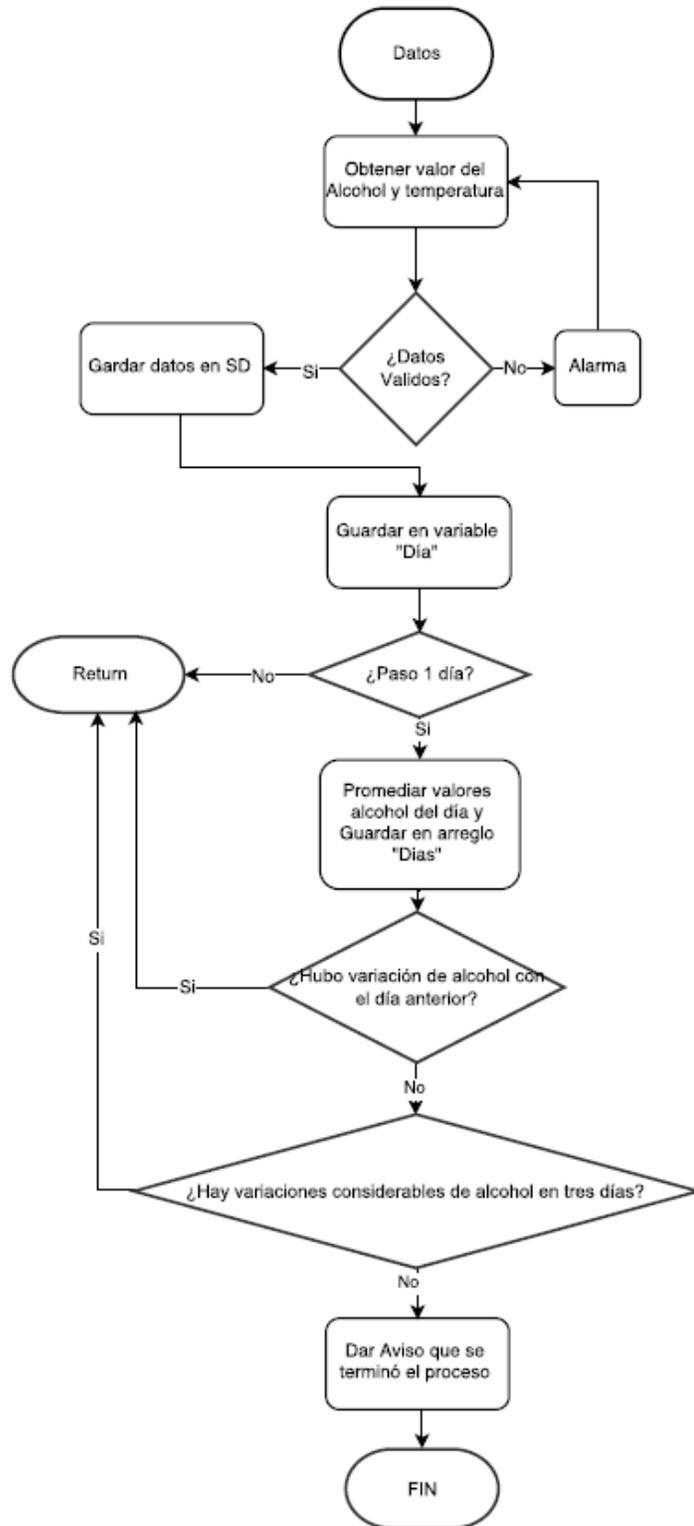
Antes que todo, esta función comienza obteniendo los valores del sensor de alcohol y temperatura. Luego, se debe verificar si los datos obtenidos son válidos o no. Esto se hace debido a que el fermentador no posee una interfaz máquina hombre, por ende, se debe alertar cualquier inconveniente del sistema. Si el dato es inválido, se arroja una alerta visual (mediante el parpadeo de una luz LED) y/o una alerta sonora (mediante un pequeño timbre o parlante). Estas alertas son diferentes dependiendo de cuál dato es inválido, dándole al usuario una idea clara del error existente. Por otro lado, si el dato es válido, se guardan los datos en la tarjeta SD.

Posteriormente, se debe conocer si los valores de etanol han llegado a su máximo y si se han mantenido estables (para dar término al proceso). Para lograr esto, primero se debe conocer si ha transcurrido un día de datos. Esto se logra debido a que se toma una muestra cada minuto, entonces cuando se tomen 1440 muestras se conoce que ha transcurrido un día. Una vez transcurrido el día, se promedian los 1440 valores y se obtiene un valor que se guarda en un arreglo "Días". Esto se repite hasta que existan tres días de datos. Cuando se cumple la condición anterior, se procede a analizar si hubo variación considerable en sus valores de etanol (porcentaje definido por el usuario) para poder discriminar si ha terminado el proceso. Si existe cambio considerable se repite el proceso hasta que exista un nuevo dato promediado en el arreglo "Días" para luego comprobar nuevamente si hubo variación considerable. Si no la hubo, el sistema da aviso visual y auditivo que la cerveza terminó su fermentación y está lista para ser embotellada, dando fin al proceso de elaboración de cerveza.

---

<sup>19</sup> Anexo N°1 (12) código datos.

Figura 18 Función Datos



Fuente: Propia (2016)

### **3.5.1.13. Alerta**

Esta función<sup>20</sup> (figura 19), se encarga de verificar el estado de los sensores de flujo y sensores de nivel con el fin de alertar al usuario, mediante la pantalla táctil, el posible error encontrado en el sistema; deteniendo su proceso.

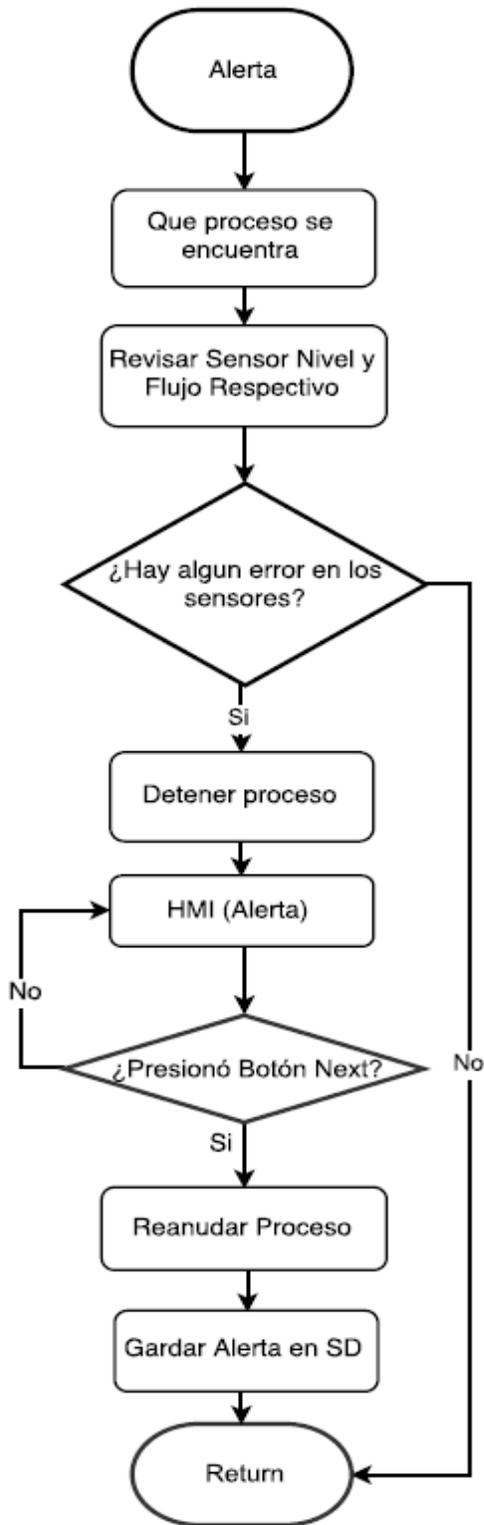
Esta función debe conocer como parámetro inicial la etapa en la cual se encuentra, debido a que se necesita conocer los sensores que debe revisar. Luego, el sistema revisa los sensores de flujo y nivel pertinentes, verificando su correcto funcionamiento. Si alguno de ellos presenta una falla, el sistema detiene el proceso en el cual se encuentre, además debe arrojar una alerta en la pantalla táctil (Anexo N°2 (15)) y guardar el registro de la alerta en la tarjeta SD. Una vez emitida la alerta el sistema espera hasta que el usuario presione el botón “*next*” para reanudar el proceso.

SOLO USO ACADÉMICO

---

<sup>20</sup> Anexo N°1 (13) código alerta.

Figura 19 Función Alerta



Fuente: Propia (2016)

## CAPITULO IV

### IMPLEMENTACION DEL SISTEMA

#### 4.1. INTEGRACION DEL SISTEMA

##### 4.1.1. Introducción

En este capítulo se detallará la implementación de todo el sistema, desde el armazón en donde se encuentran los recipientes, mangueras, etc., hasta los actuadores, sensores, controladores, entre otras cosas.

También se especificarán los sensores, actuadores, módulos utilizados, conociendo sus características principales, modo de funcionamiento, entre otras cosas.

Además, se mostrarán las diferentes pruebas realizadas para las distintas etapas del proceso. Todo esto, con el fin de determinar un correcto funcionamiento del sistema y del diseño descrito en los capítulos anteriores.

##### 4.1.2. Estructura del Sistema

En la figura 20 se muestra un diagrama el cual ilustra todos los componentes del sistema y su disposición general con su respectiva leyenda.

Analicemos este sistema por partes. El recipiente uno denominado "*Tanque Agua*", se compone de un calentador y sus dos sensores de nivel colocados al interior (uno superior y otro inferior) (Anexo N°3 (1)). A la salida del recipiente se conecta una manguera la cual está conectada a la entrada de la bomba de recirculación. A la salida de esta bomba, se conectó el sensor de temperatura, con el objetivo de obtener la temperatura al momento de recircular el líquido por medio de una copla en "*T*" de cobre y una copla que posee hilo interno de media pulgada. Posteriormente, el flujo se divide en dos canales, cada uno de ellos con una válvula de solenoide y un sensor de flujo, lo cual dependiendo de cuál válvula esté activada, dirige el flujo hacia el primer recipiente o al segundo recipiente según sea el caso (Anexo N°3 (2)).

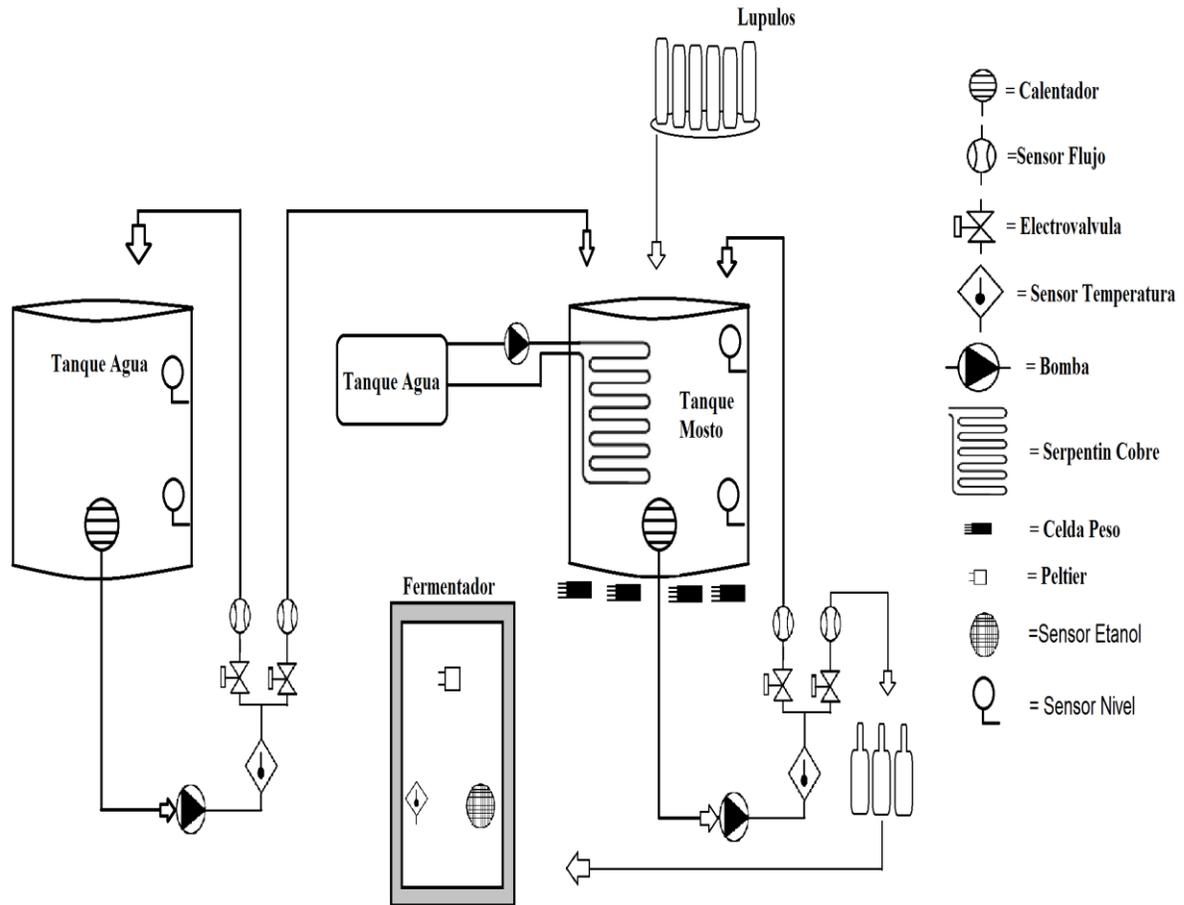
El segundo recipiente, denominado "*Tanque Mosto*", se compone de los mismos elementos del recipiente uno, con la diferencia de tres elementos extras. El primer elemento extra, es el alimentador de lúpulos (Anexo N°3 (3)). Éste se instala en el

borde del recipiente, quedando en posición ideal para botar el lúpulo al interior del recipiente. El segundo elemento extra es el serpentín de cobre (Anexo N°3 (4)). Este dispositivo es removible y se instala al momento de comenzar el hervido o cocción para su posterior uso. El tercer elemento extra son los sensores de peso, los cuales se utilizan en la etapa de macerado y hervido. Éstos deben estar montados por sus dos extremos. El primer extremo está sujeto al armazón principal de metal, quedando en una posición fija. El otro extremo está sujeto a una placa de madera, quedando suspendido (Anexo N°3 (5)). Gracias a su montaje el sensor de peso logra flexionarse, logrando su correcto funcionamiento.

El fermentador se compone de un cajón de madera terciada de 15 [mm] el cual tiene en sus seis paredes internas un “*plumavit*” de 50 [mm] de espesor para ayudar con la aislación (Anexo N°3 (6A)). Además, está recubierto con un aislante térmico aluminizado el cual ayuda aún más con la aislación (Anexo N°3 (6B)). En una de sus paredes se encuentra el “*peltier*” con sus respectivos disipadores de aluminio. Igualmente, están los sensores de temperatura y alcohol. El sensor de temperatura se encuentra montado al interior del cajón con el propósito de medir la temperatura en el interior. En cuanto al sensor de alcohol, se encuentra montado sobre el “*airlock*” con el uso de una pieza impresa en 3D para este proyecto (Anexo N°3 (7)). Esta montura permite el paso directo de los gases de fermentación por el sensor de alcohol, logrando su correcto funcionamiento.

El prototipo del sistema se encuentra montado sobre un “carrito” fabricado de perfil de fierro y con ruedas para su cómodo transporte (Anexo N°3 (8)).

Figura 20 Estructura del Sistema

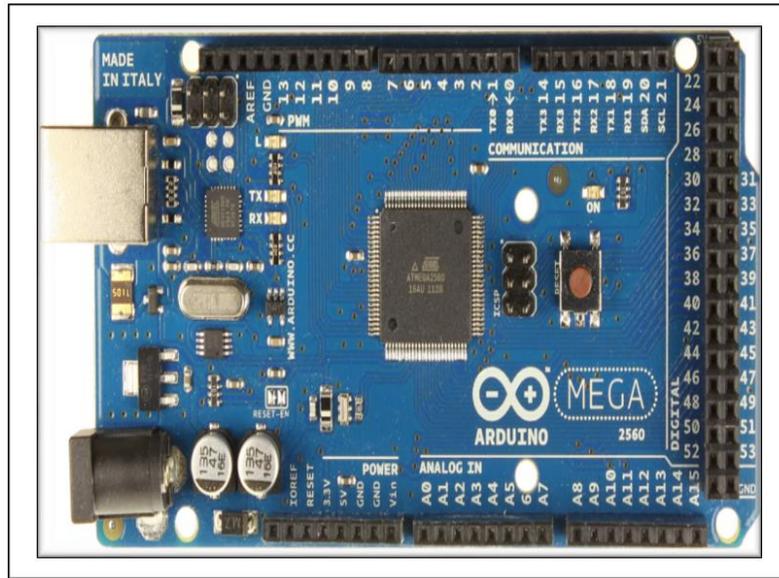


Fuente: Propia (2016)

#### 4.1.3. Controlador

Los dos controladores utilizados se encargan de controlar y monitorear los sensores y actuadores a lo largo de todo el proceso. Se optó por utilizar el *Arduino Mega 2560* (figura 21) como controlador uno y dos. Este está construido con en el microcontrolador ATmega2560, el cual pertenece a la compañía *Arduino* de hardware libre. Este microcontrolador posee 54 pines de entradas/salidas digitales, 16 entradas análogas, cuatro puertos seriales, entre otras características, suficientes para realizar todos los procesos diseñados.

Figura 21 Controlador Arduino Mega 2560



Fuente: Propia (2016)

#### 4.1.4. Sensores

##### 4.1.4.1. Temperatura

Se utilizó el sensor DS18B20 (figura 22), el cual es un sensor digital que utiliza el protocolo 1-Wire. Éste posee diversas características siendo la principal la posibilidad de conectar múltiples sensores (compatibles con este protocolo) en un solo bus de datos ya que cada DS18B20 posee un único número de serie. Además, si se desea, permite energizarlos mediante este mismo bus de datos, eliminando la necesidad de una fuente externa (Anexo N°4 (1)). También este sensor viene con un encapsulado TO-92 el cual se encuentra dentro de un cilindro de acero inoxidable y unido con los cables mediante el uso de termo retráctil, logrando su impermeabilidad con el fin de poder ser sumergido en diferentes líquidos.

Finalmente, se escogió este sensor porque cumple con todas las características necesarias para este proyecto, como lo son: rangos de temperatura, tiempo de respuesta y precisión.

Figura 22 Sensor DS18B20



Fuente: Propia (2016)

#### 4.1.4.2. Peso

Se utilizó una celda de peso la cual es capaz de convertir la carga o fuerza aplicada en una señal eléctrica. Se utilizó específicamente el modelo *FZ1439* (Anexo N°4 (2), figura 23) el cual soporta una carga máxima de 20 [Kg] por celda y posee agujeros para su montaje.

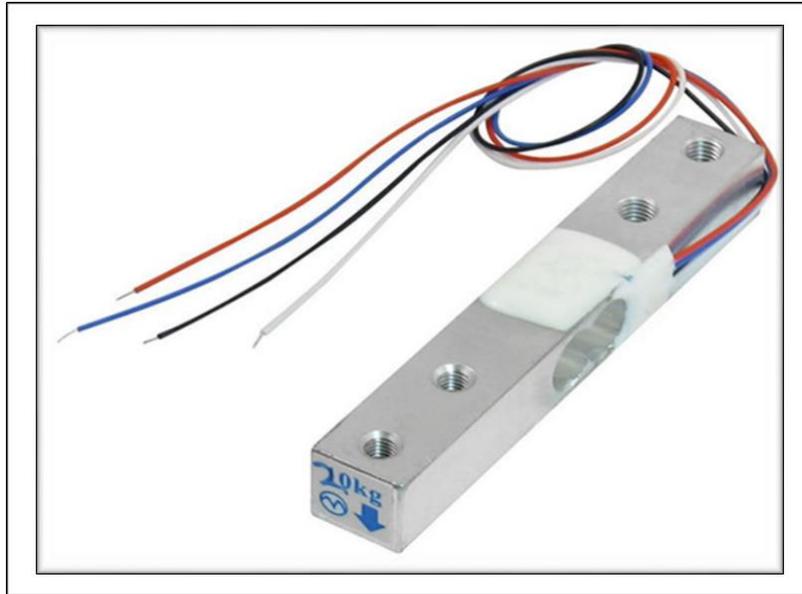
Este sensor posee cuatro cables, configurados como un puente *wheatstone*<sup>21</sup>. A dos de estos cables se le aplica un voltaje de excitación y los otros dos restantes se les debe medir el voltaje. En estado de equilibrio, sin ninguna fuerza aplicada al sensor, el voltaje de salida (del segundo par de cables) debe ser cercano a cero, ya que los valores de las cuatro resistencias son similares. En caso contrario, si se aplica una fuerza al sensor, genera una variación del valor de una o más resistencias, ocasionando un pequeño cambio en el voltaje de salida de alrededor de unos *millivolts*.

---

<sup>21</sup> La configuración de cuatro resistencias que forman un circuito cerrado. Se utiliza para medir resistencias desconocidas mediante el equilibrio de los brazos del puente.

Esta señal se debe acondicionar, ya que es muy pequeña para poder leerla con el controlador utilizado, el cual posee un rango de 0 a 5 [V] en su entrada análoga. Debido a esto, se utilizó el conversor análogo digital *Hx711* (Anexo N°4 (3)) el cual está especialmente diseñado para trabajar con celdas de carga de este tipo y ofrecer una salida de voltaje, acondicionada para el controlador.

Figura 23 Celda de Peso FZ1439



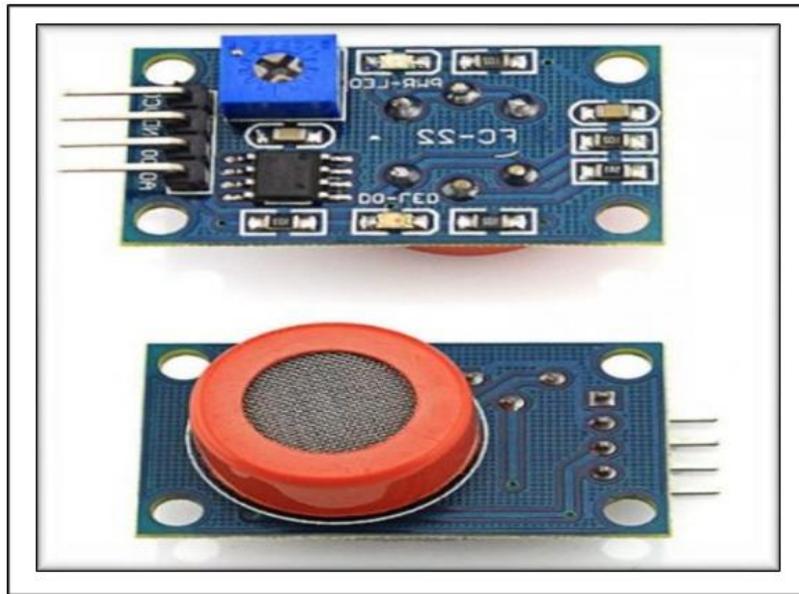
Fuente: Propia (2016)

#### 4.1.4.3. Alcohol

El sensor de alcohol utilizado es un módulo que utiliza el sensor de gas MQ3 (figura 24), el cual es altamente sensible a los gases de alcohol (Anexo N°4 (4)). El sensor posee un calentador interno el cual provee las condiciones necesarias para el correcto funcionamiento del sensor. Éste debe estar previamente calentado por más de 24 horas para tomar datos válidos.

El modo de funcionamiento es principalmente resistivo. Al estar en presencia de un gas en el aire, el sensor reduce su resistencia proporcionalmente a la cantidad de gas presente. Debido a que el módulo está configurado como un divisor de voltaje, al variar la resistencia interna del MQ3, varía la salida del divisor de voltaje. Este voltaje es amplificado, acondicionando la señal para la entrada análoga del controlador.

Figura 24 MQ3 sensor



Fuente: Propia (2016)

#### 4.1.4.4. Flujo

El sensor de flujo o fluxómetro utilizado es el YF-S201 (Anexo N°4 (5)) (figura 25), el cual puede medir entre 1 y 30 [L/min] de caudal. Este sensor posee una hélice como rotor el cual tiene integrado un imán y un sensor de *efecto hall* en su estator. Esta hélice rota en el momento que fluye un líquido entre sus aberturas, provocando que el sensor genere pulsos eléctricos por cada rotación de la hélice. Estos pulsos son detectados por el controlador el cual, según la frecuencia de los pulsos, determina el caudal del líquido.

Figura 25 Fluxómetro YF-S201



Fuente: Propia (2016)

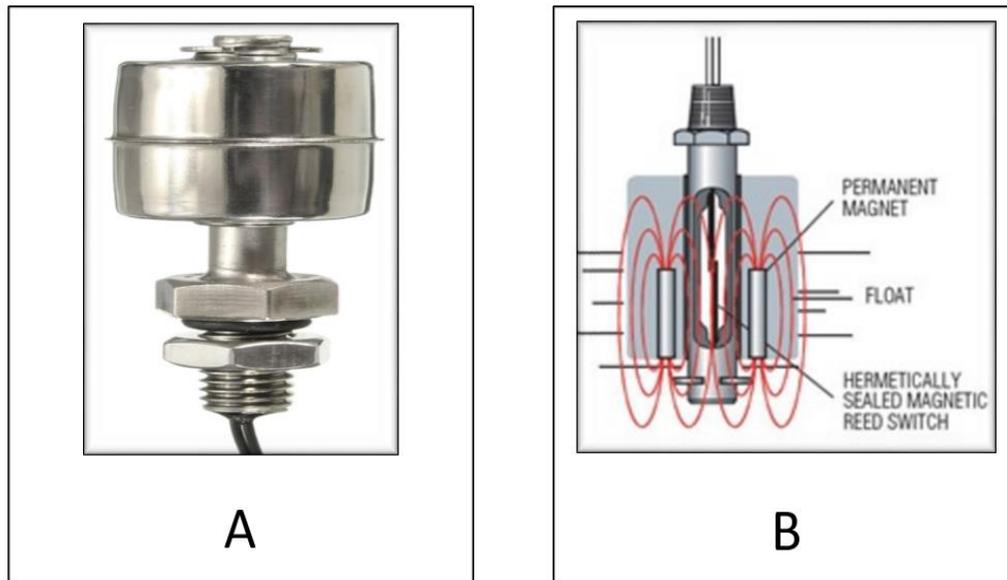
#### 4.1.4.5. Nivel

El sensor de nivel utilizado es el que se muestra en la figura 26 (A), el cual está fabricado de acero inoxidable y posee una rosca en su parte inferior para su montaje (Anexo N°4 (6)).

Este sensor es de tipo flotador, el cual, está compuesto principalmente de dos partes, el flotador cilíndrico que posee un imán en su interior y la base que posee un interruptor de lengüeta (figura 26 (B)).

El modo de funcionamiento es sencillo. Al momento de subir el líquido causa que el flotador suba con él, provocando que el campo magnético del imán, atraiga la lengüeta cerrando el circuito. De esta forma puede ser detectado por el controlador con el fin de realizar la acción pertinente.

Figura 26 Sensor Nivel Exterior (A), Sensor Nivel Interior (B)



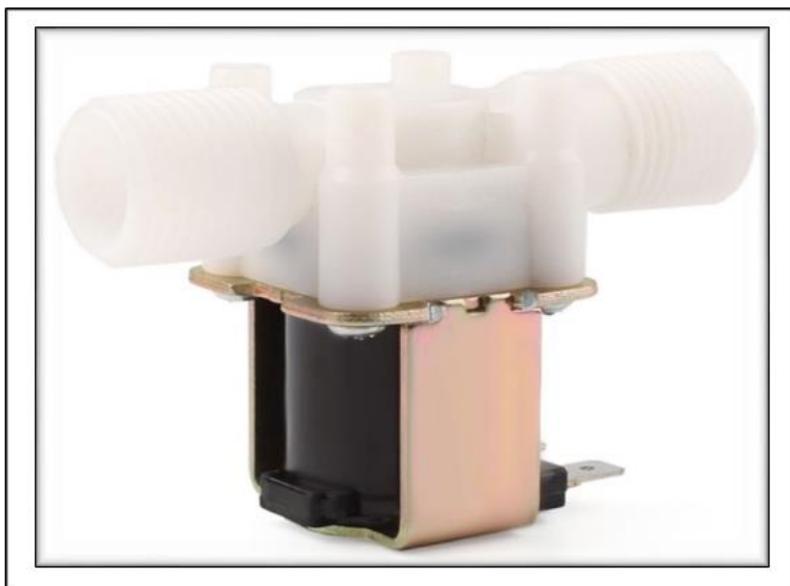
Fuente: Propia (2016)

#### 4.1.5. Actuadores

##### 4.1.5.1. Válvula Solenoide

La válvula de solenoide es un dispositivo operado eléctricamente y utilizado para controlar el flujo de líquidos. La válvula utilizada es la ZE-4F180 (Anexo N°4 (7)) mostrada en la figura 27. Ésta, es normalmente cerrada, esto quiere decir que se cierra por la acción de un resorte sin energizar la bobina, cortando el flujo del líquido. Para abrir la válvula se debe energizar la bobina, provocando un campo magnético. Esto provoca un movimiento del émbolo (fabricado de material magnético) hacia el centro de la bobina, logrando levantarlo lo cual abre el sello e induce un flujo de líquido entre sus extremos.

Figura 27 Válvula Solenoide



Fuente: Propia (2016)

#### 4.1.5.2. Bombas

La bomba utilizada en ambos recipientes es el de la figura 28 (A). Utiliza 12 [V] corriente continua y es utilizada para mover el agua o el mosto entre los recipientes o recircular a un mismo recipiente. Ésta soporta fluidos de hasta 110 [°C], ideales para este proyecto (Anexo N°4 (8)).

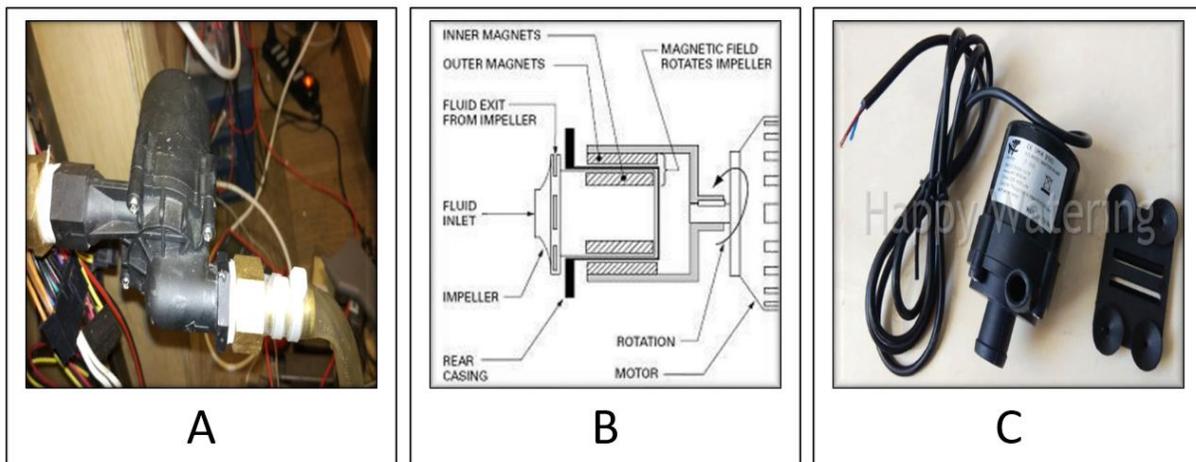
El sistema de propulsión de la hélice es de categoría alimentaria, lo cual fue su característica principal de elección. Éste usa un acople magnético el cual reemplaza la conexión física con la hélice por medio de un campo magnético.

Como se puede apreciar en la figura 28 (B), el rotor del motor está acoplado a una carcasa de plástico (color gris sólido) la cual posee imanes en su interior. Cuando el motor se enciende, comienzan a girar los imanes de la carcasa, generando un campo magnético el cual afecta a los imanes internos que se encuentran acoplados con la hélice. Esto genera rotación de la hélice y por ende genera propulsión del líquido.

Al estar aislado las partes del motor con la hélice, proporciona una superficie más higiénica evitando la producción de bacterias y/o elementos no deseados que arruinen la calidad de la cerveza.

Por último, la bomba utilizada en la etapa de enfriamiento se muestra en la figura 28 (C). Esta funciona con 12 [V] corriente continua y es completamente sumergible, siendo un candidato perfecto para la aplicación (Anexo N°4 (9)).

Figura 28 Bomba Alimentaria (A), Bomba Alimentaria Interior (B), Bomba Enfriamiento (C)



Fuente: Propia (2016)

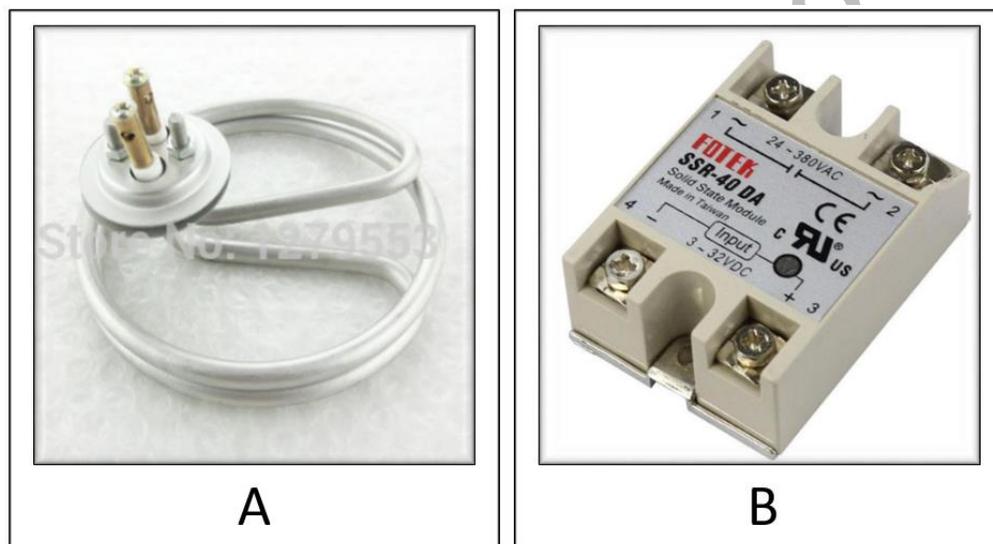
#### 4.1.5.3. Calentadores

El elemento calentador es utilizado en gran variedad de aplicaciones desde la industria alimenticia (hervidores eléctricos) hasta la agropecuaria, calentando el agua de los tanques.

Se optó por utilizar uno de acero inoxidable el cual se conecta a 220 [V], corriente alterna figura 29 (A). Posee dos pernos de montaje con sus respectivos sellos con el fin de asegurar su perfecta impermeabilidad. El elemento calentador opera mediante la conversión de la electricidad suministrada por sus bornes en energía calórica. Esto lo logra ya que, al ser un dispositivo resistivo, se opone al flujo de la corriente generando calor (Anexo N°4 (10)).

Debido a que es un dispositivo que posee alimentación por corriente alterna no se puede controlar directamente con el controlador utilizado. Por ende, se debe ocupar un elemento el cual soporte este tipo de cargas. Este elemento es el denominado SSR (Anexo N°4 (11)) (Solid State Relay, por sus siglas en ingles), el cual, mediante el uso de transistores, tiristores o *triacs*, controlan cargas de potencia a partir de señales de control de bajo voltaje. En otras palabras, mediante el uso del controlador se aplica una señal a la entrada de control del SSR con tal de controlar una carga de potencia como lo es el elemento calentador (figura 29 (B)).

Figura 29 Elemento Calentador (A), SSR (B)



Fuente: Propia (2016)

#### 4.1.5.4. Alimentador de Lúpulos

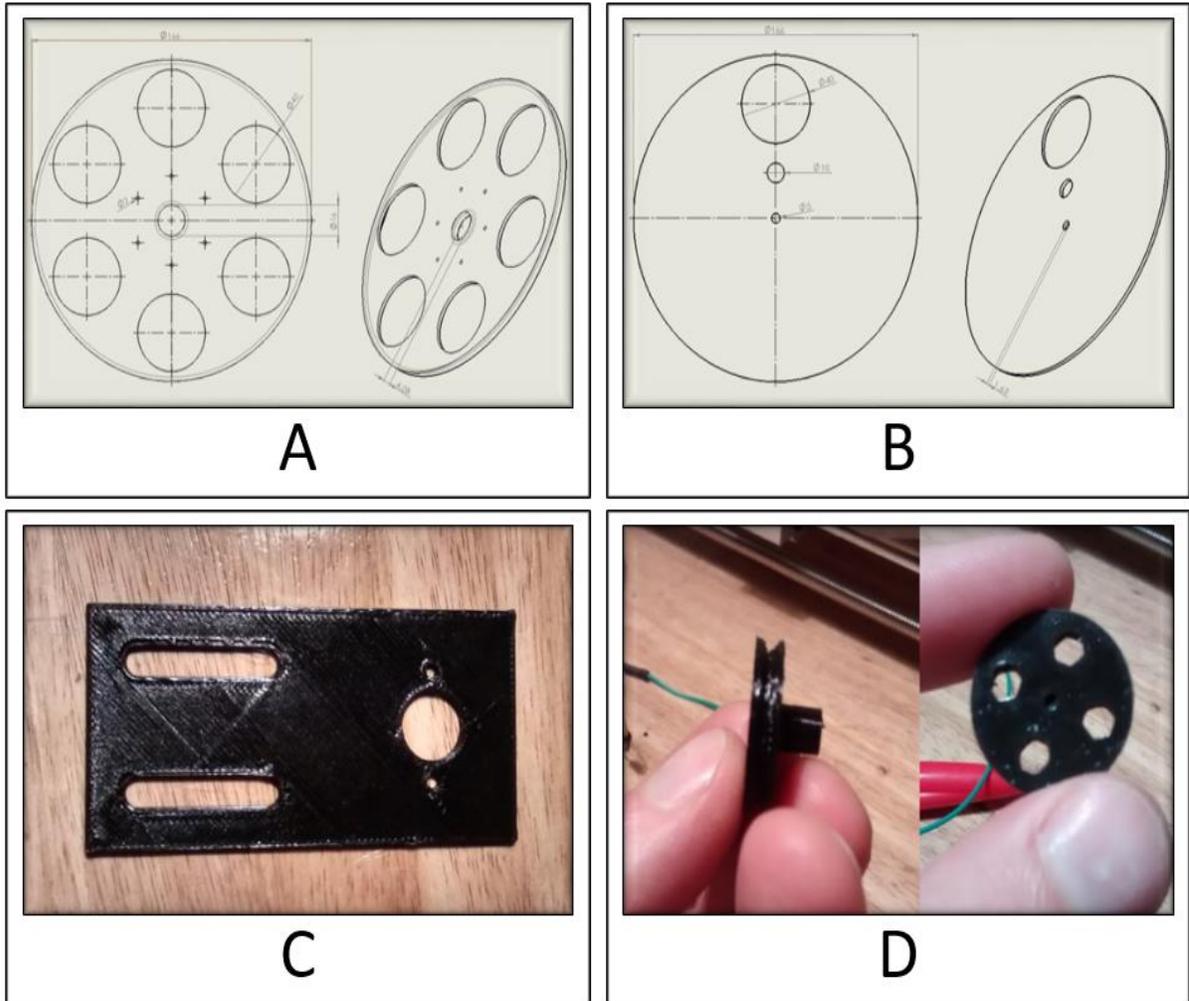
Este actuador, fue diseñado y fabricado exclusivamente para este proyecto, utilizando referencias de uno fabricado por la compañía *Brew-Boss*. Éste consiste de dos discos de madera redondos. El disco superior (figura 30 (A)) posee seis orificios donde se insertan unos tubos de PVC los cuales se usarán para almacenar los lúpulos y la levadura al momento de la elaboración de la cerveza. Además, este disco posee un agujero central el cual se insertará un rodamiento para generar un movimiento rotatorio fluido. Por último, posee seis agujeros más, los cuales servirán para conocer la posición del lúpulo que se encuentra. El disco inferior (figura 30(B)) posee un orificio

del mismo tamaño del disco anterior con el objetivo de que al momento que coincidan los dos orificios, caiga el respectivo contenido. El orificio de 10 [mm] es donde se colocará un interruptor de final de carrera con el objetivo de que, al momento que coincida con los agujeros de 2 [mm] del plato superior, el interruptor se levante y detecte la posición. Finalmente, el orificio central es para colocar el eje de rotación y fijarlo al rodamiento del primer disco.

Una vez ensamblado lo anterior, se debe diseñar e implementar el mecanismo que rota el dispositivo. Se optó por realizarlo con un motor paso a paso, el cual mediante un controlador se puede controlar fácilmente. Este motor está fijado al disco inferior mediante un soporte impreso por una impresora 3D (figura 30 (C)) y posee en su rotor una polea igualmente impresa con un pequeño sacado a lo largo de su perímetro (figura 30 (D)). Este sacado se realizó ya que este motor será acoplado al disco superior mediante una correa.

En cuanto al funcionamiento, éste es relativamente sencillo. Al momento que se desee botar el contenido, el controlador manda la señal al controlador del motor provocando que gire. Esto lo hace hasta que el interruptor fin de carrera detecte que se ha levantado (cuando los agujeros de los platos inferior y superior coinciden). Es allí donde el controlador deja de enviar la señal al controlador del motor, deteniendo su giro.

Figura 30 Disco Superior (A), Disco Inferior (B), Soporte Motor (C), Polea Motor (D)



Fuente: Propia (2016)

#### 4.1.6. HMI

El HMI (Human Machine Interface, por sus siglas en inglés) es una interfaz de usuario que se usa para la interacción entre el humano y la máquina. El dispositivo que se seleccionó es el *NX4832T035* (figura 31). Este dispositivo es fabricado por la compañía *Itead*, empresa de origen chino, que se encarga de diseñar y manufacturar dispositivos electrónicos (Anexo N°4 (12)).

Figura 31 HMI



Fuente: Propia (2016)

El NX4832T035 es una pantalla táctil de 3.5", que en conjunto con el software *Nextion Editor* son una excelente herramienta para realizar interfaces sencillas, rápidas de diseñar y utilizar. En este software se programa el NX4832T035 por medio de un cable USB, el cual se conecta al computador. El software es uso intuitivo pudiendo implementar botones, deslizadores, fotos, entre otras cosas, asignándoles acciones o parámetros a los objetos agregados para su posterior uso con el controlador. Una vez satisfecho con la interfaz diseñada, se compila y carga la interfaz a la pantalla. Hecho esto, está lista para su uso.

Para usarlo con el controlador seleccionado, se debe conectar a uno de los puertos seriales del controlador. Hecho esto, se debe agregar elementos al código como librerías, variables, entre otras cosas, para poder interactuar con éste. Esta interacción es bidireccional (enviando y recibiendo datos), pudiendo tratarse de diferentes variables como la temperatura actual de cada recipiente, mensajes de alerta, estado de los sensores, entre otras cosas.

## 4.2. PRUEBAS

### 4.2.1. Introducción

Para llevar a cabo todo lo diseñado en este proyecto, se deben realizar diversas pruebas las cuales avalen los diseños. Se realizaron cinco pruebas con el fin de comprobar métodos, teorías y tecnologías planteadas. En cada una de las cinco pruebas realizadas se definió sus respectivas herramientas y metodologías para llevarlas a cabo de la mejor manera.

### 4.2.2. Desarrollo

La primera prueba permite en definir los parámetros del controlador PI del recipiente con el fin de poder alcanzar la temperatura deseada de forma precisa, contrarrestando las perturbaciones externas. Debido a que los dos recipientes poseen los mismos actuadores y sensores, se realizó la prueba con uno solo, asumiendo que los valores obtenidos del controlador PI, funcionaran correctamente en ambos.

Para lograr esto, se optó por utilizar *Matlab*. Ésta es una herramienta de software matemático con un lenguaje de programación propio utilizado en un gran número de aplicaciones, ya que posee diferentes características para diseñar y analizar diferentes sistemas. Matlab posee una aplicación denominada “*PID Tuner*”, la cual obtiene los parámetros del controlador mediante el uso de los datos de entrada y salida del sistema (Turevskiy, 2012).

Se comenzó con el llenado de un recipiente con agua a temperatura ambiente. Luego se inició la obtención de los valores de temperatura (señal de salida) cada un segundo, manteniendo el registro en la tarjeta SD. Posteriormente, se activó el calentador en un tiempo conocido con una señal de tipo escalón (señal de entrada), activándolo a su máxima potencia. A continuación, se mantuvo el calentador y el registro de la temperatura hasta que la temperatura alcanzó su máximo estable (ebullición del agua) con el fin de conocer el comportamiento de la planta (figura 32 (A)).

Una vez conocido la señal de entrada y salida de la planta, se procede a utilizar el *PID Tuner* para identificar la planta. Ya identificada la planta (figura 32 (B)) con los cálculos pertinentes de Matlab, se procede a determinar los parámetros del controlador según

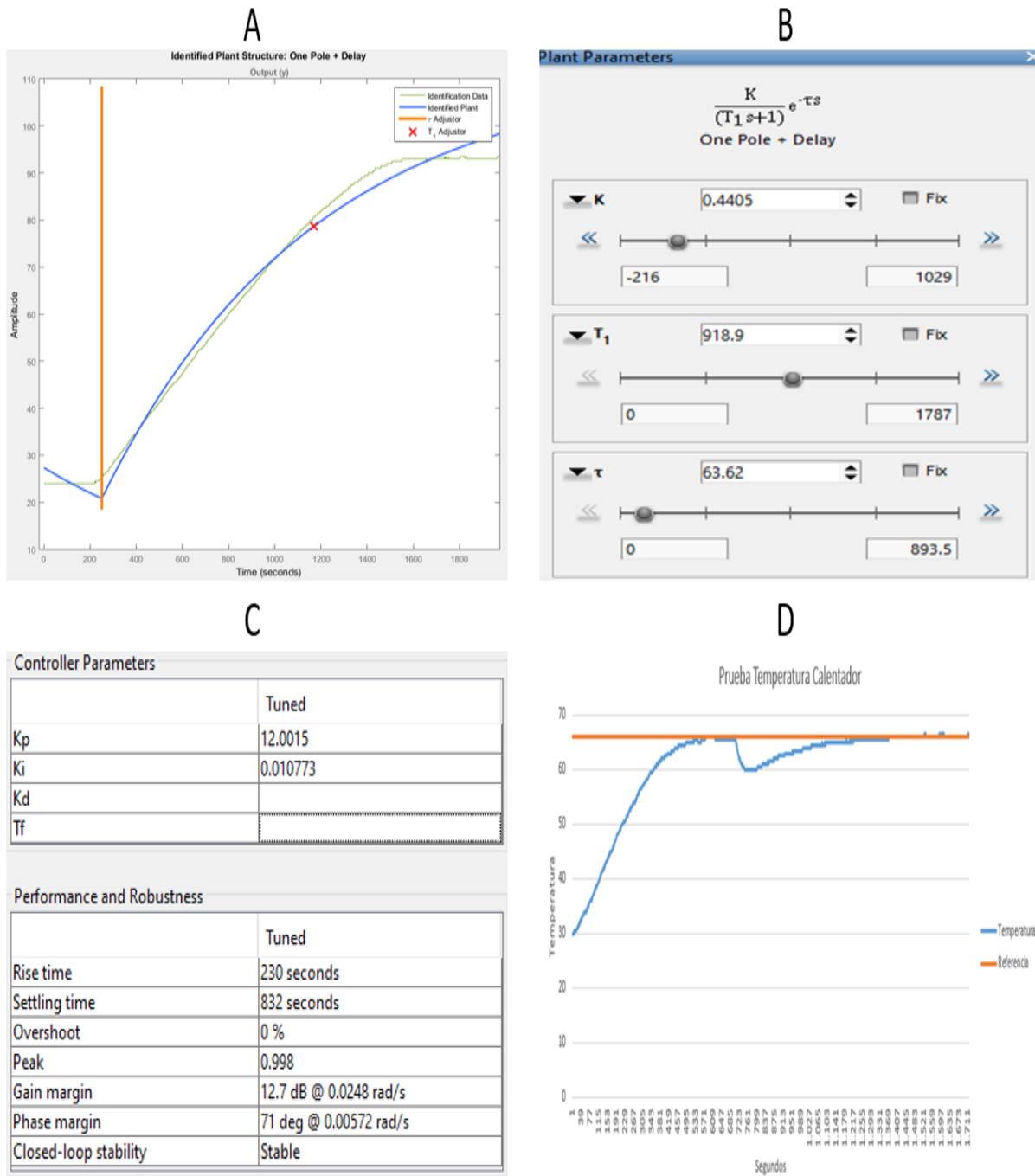
el rendimiento esperado. Esto se logra ajustando la rapidez de respuesta y la robustez del controlador requerido. Definido lo anterior la aplicación entrega los parámetros del controlador diseñado, para su posterior implementación (figura 32 (C)).

Para comprobar la efectividad de los parámetros del controlador, se realizó una prueba con los parámetros obtenidos. La prueba consistió en definir un valor arbitrario como temperatura de referencia, el cual fue de 66 [°C]. Luego se llenó el recipiente con agua a temperatura ambiente y se activó el control PI con el fin de alcanzar la temperatura de referencia.

Los resultados se pueden apreciar en la figura 32 (D), pudiendo observar que el sistema logró alcanzar la temperatura de referencia, incluso después de una perturbación (arrojar agua fría al recipiente), logrando un error máximo de 0.76% en su periodo estacionario.

SOLO USO ACADÉMICO

Figura 32 Comportamiento de la Planta (A), Función de Transferencia de la Planta (B), Parámetros del Controlador de los Elementos Calentadores (C), Prueba realizada al Calentador (D)



Fuente: Propia (2016)

La segunda prueba permite definir los parámetros del controlador PI para el fermentador. Para ello se realizó el mismo procedimiento mencionado anteriormente pero utilizando un actuador diferente (*peltier*). Realizado lo anterior se obtuvo los parámetros del controlador para el fermentador (figura 33 (A)).

Nuevamente para comprobar la efectividad de los parámetros obtenidos del controlador, se realizó una prueba. La prueba consistió en definir la temperatura de referencia de 23 [°C] y se activó el control PI hasta que el sistema alcance la temperatura de referencia. Se puede apreciar en la figura 33 (B) como la temperatura alcanzó la referencia logrando un error máximo en su periodo estacionario de 0.83%.

Figura 33 Parámetros del Controlador del Fermentador (A), Prueba Realizada al Fermentador (B)

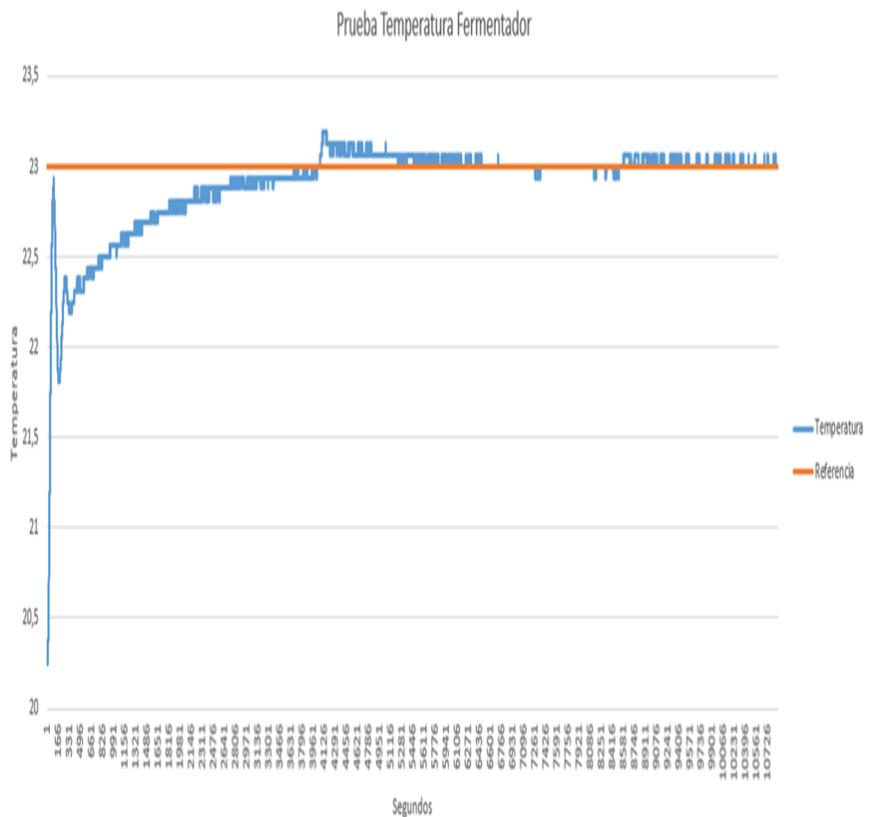
A

Controller Parameters	
	Tuned
Kp	64.9474
Ki	0.30544
Kd	
Tf	

Performance and Robustness	
	Tuned
Rise time	52.3 seconds
Settling time	93.2 seconds
Overshoot	0%
Peak	1
Gain margin	Inf dB @ NaN rad/s
Phase margin	90 deg @ 0.042 rad/s
Closed-loop stability	Stable

B



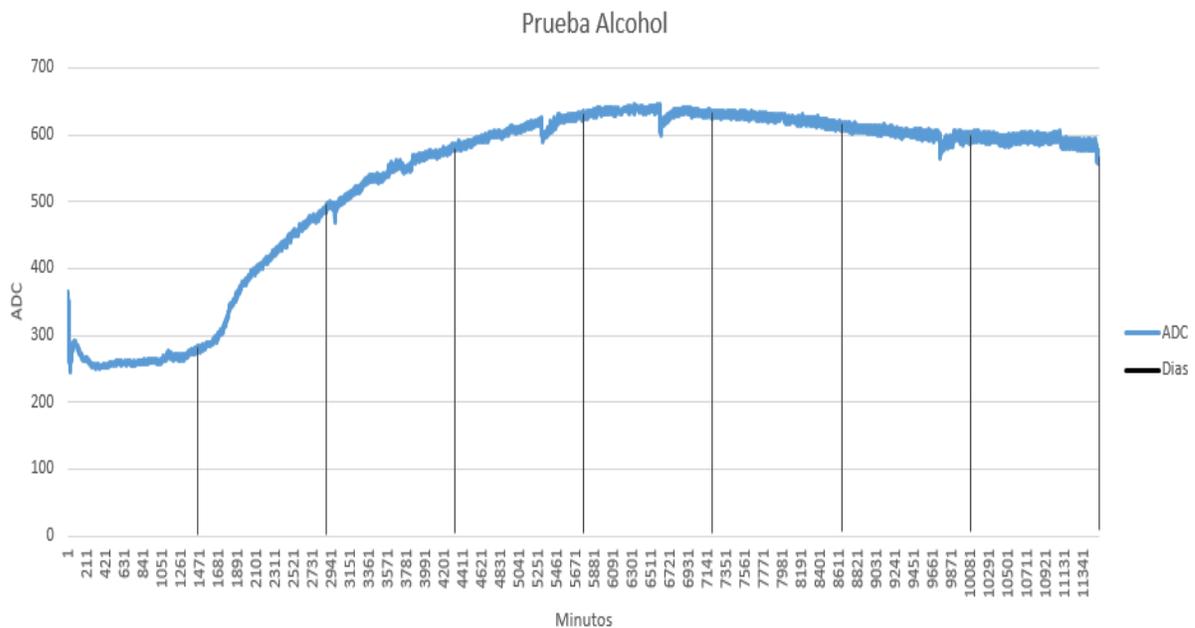
Fuente: Propia (2016)

La tercera prueba consiste en monitorear la cantidad de alcohol producido por la fermentación de la cerveza y comprobar que el sistema da aviso al momento en que ya llegó a su máximo estable con el fin de terminar el proceso de fermentación.

Con la función “*Datos*” descrita en el capítulo anterior, la cual guardaba los datos del sensor de alcohol cada un minuto y detenía el proceso al momento que no presentaba variaciones considerables en su valor promediado. Se puede apreciar en la figura 34 los datos obtenidos en ocho días. Se ve como en los últimos tres días no hubo variación apreciable entre ellos (Anexo N°5 (1)). Por lo tanto, el sistema dio aviso de término de la etapa de fermentación.

Además, se puede observar en la figura 34 unas caídas abruptas en los valores del sensor. Esto se debe a, que se detuvo el sistema con el fin de retirar la tarjeta SD y sacar los datos de ella. Por ende, esto causó que la resistencia del sensor se enfriara, alterando los datos. Luego, transcurrido aproximadamente cuatro horas, el sensor estabiliza sus valores.

Figura 34 Prueba Sensor de Alcohol



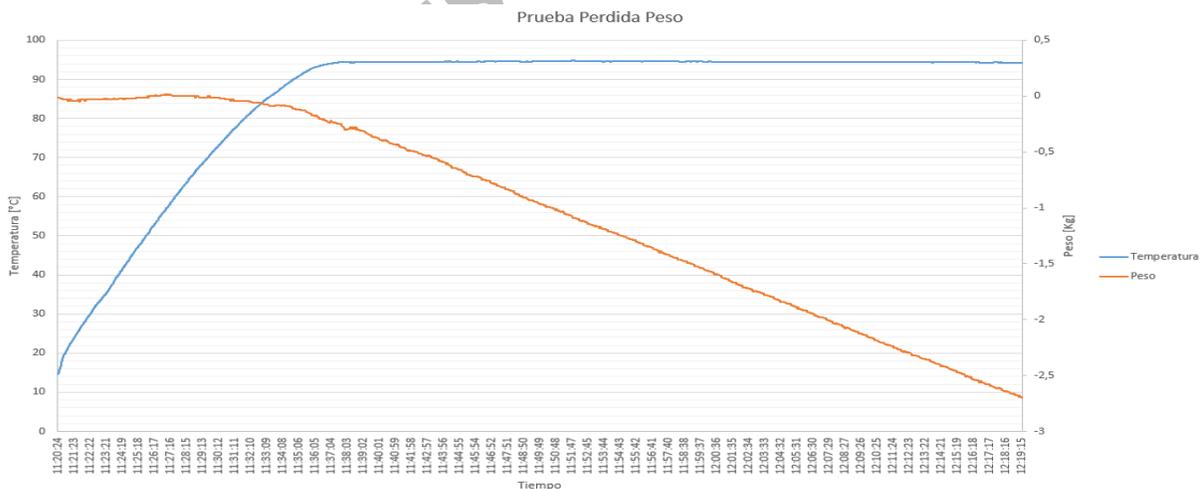
Fuente: Propia (2016)

La cuarta prueba es cuantificar la cantidad de agua que se pierde en la etapa de cocción o hervido con el fin de poder estimar de mejor manera el agua requerida en la etapa de “Lavado”.

Esto se logró colocando agua a temperatura ambiente en el segundo recipiente, reiniciando el valor de la pesa una vez lleno el recipiente. A continuación, se activó el calentador a su máxima potencia con tal de llegar al punto de ebullición del agua. Paralelamente, se mantuvo guardando datos cada un segundo para su posterior análisis. Realizado lo anterior, se pudo graficar y posteriormente analizar los datos.

Se puede apreciar en la figura 35 los tres datos relevantes en esta prueba: temperatura, peso, y tiempo. Se puede observar que la pérdida de peso es relativamente lineal pudiendo extrapolar la cantidad de agua perdida de forma directa mediante la siguiente ecuación  $y = -0.0013x + 0.4765$ <sup>22</sup>. Por lo tanto, la pérdida de agua es de aproximadamente **4.2 [L/h]** y debido a esta estimación, el sistema puede calcular y compensar esa pérdida de agua en la etapa de lavado con el fin de mantener la relación de malta/agua deseada.

Figura 35 Prueba Perdida de Agua mediante el monitoreo de su peso



Fuente: Propia (2016)

<sup>22</sup> Siendo  $y$ = peso perdido en litros;  $x$ = segundos transcurridos

La quinta y última prueba consiste en probar el sistema completo con tal de detectar posibles fallas y verificar el funcionamiento correcto de todas las funciones (excluyendo la etapa de fermentación, ya que, ya se realizaron las pruebas).

Iniciando esta prueba, en la figura 36 se muestra el protocolo establecido al momento de guardar los datos en la tarjeta SD. Se pueden observar los parámetros definidos para esta prueba en la fila 1 y 2, respectivamente.

Figura 36 Protocolo establecido al momento de guardar los datos

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Temp Macera	Tiempo Macer	Temp Hervido	Tiempo Hervid	Tiempo Lupulc	Tiempo Lupulc	Tiempo Lupulc	Cant Malta	Relacion Malt	Cantidad de M	Hora	Fecha	
2	65	15	100	15	15	10	5	1026.10	1	1026	9:30:14	02-08-2016	
3	Setpoint 1	Temperatura	Output 1	Setpoint 2	Temperatura	Output 2	Tiempo Faltan	Tiempo Proce	Hora	Peso Mosto	Cantidad Agua	Etapas	Alerta
4	65	10.56	255.00				0	0	9:30:25			Calentando	OK
5	65	10.56	255.00				0	0	9:30:36			Calentando	OK
6	65	10.69	255.00				0	0	9:30:48			Calentando	OK
7	65	11.00	255.00				0	0	9:30:59			Calentando	OK
8	65	11.38	255.00				0	0	9:31:11			Calentando	OK
9	65	11.88	255.00				0	0	9:31:23			Calentando	OK
10	65	12.38	255.00				0	0	9:31:34			Calentando	OK
11	65	12.75	255.00				0	0	9:31:46			Calentando	OK
12	65	13.19	255.00				0	0	9:31:58			Calentando	OK
13	65	13.63	255.00				0	0	9:32:09			Calentando	OK
14	65	14.06	255.00				0	0	9:32:21			Calentando	F1-SN.Sup
15	65	14.44	255.00				0	0	9:32:35			Calentando	OK
16	65	14.88	255.00				0	0	9:32:47			Calentando	OK
17	65	15.31	255.00				0	0	9:32:58			Calentando	OK
18	65	15.75	255.00				0	0	9:33:10			Calentando	OK
19	65	16.19	255.00				0	0	9:33:21			Calentando	OK

Fuente: Propia (2016)

Además, se observa en la fila 3 el orden de las variables de interés. En las columnas A, B y C se muestran las variables de referencia, temperatura y salida del controlador PI del primer recipiente, seguido por las mismas variables del segundo recipiente. Luego se encuentra el tiempo faltante y el tiempo total de cada etapa (columna G y H), seguido por la hora del RTC (columna I), peso del segundo recipiente (columna J), cantidad de agua (columna K), etapa en la cual se encuentra (columna L). Finalmente, la posible alerta (columna M) la cual se ve que el sistema no presenta fallas excepto en **M14** enviando un mensaje de alerta “F1-SN.Sup” significando que el fondo 1 (primer recipiente) presenta fallas en el sensor de nivel superior.

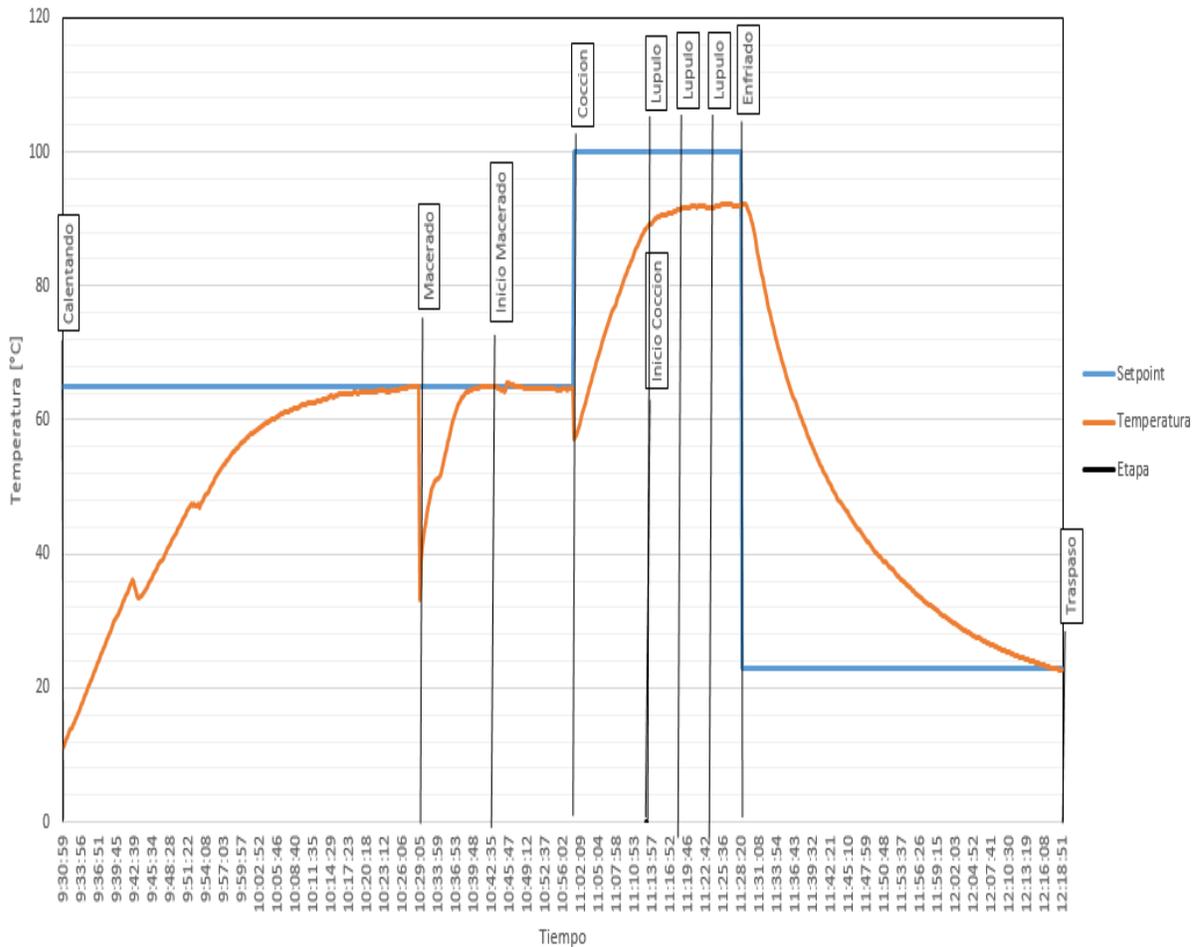
Por último, se muestra en la figura 37 la variación de temperatura a lo largo de todo el proceso, demostrando que el controlador PI de ambos recipientes, funcionan correctamente.

Luego de comenzada la etapa de “Macerado”, se observa una caída abrupta de temperatura. Fue debido a la pérdida de calor al momento de traspasar de un recipiente caliente al segundo recipiente a temperatura ambiente. Independiente de esta gran perturbación, de aproximadamente 32 [°C], el sistema fue capaz de contrarrestarlo y volver a su temperatura de referencia.

Figura 37 Prueba del Sistema Completo



Prueba Sistema Completo



Fuente: Propia (2016)

## CAPITULO V

### ANALISIS DE INVERSIONES Y COSTOS

La fabricación de un prototipo es un proceso extenso el cual involucra varias etapas. Cada una de ellas posee costos asociados e incluso, en algunas etapas, diversas iteraciones las cuales reflejan costos extras. Estas iteraciones ocurren en diversas instancias, como lo puede ser en la etapa de investigación, diferentes testeos de sensores, actuadores, estructura, diseño, como en otras cosas, lo que conlleva a costos extremadamente altos.

Debido a esto, la venta de un producto comercial debe tomar en consideración todos estos gastos asociados ya sea monetario, horas hombre, marketing, entre otras y de alguna forma u otra reflejar éstos, en el precio de venta con el fin de solventar los costos que conlleva la fabricación de un producto.

No obstante, para fines de este proyecto de título, solo se dará a conocer los costos asociados de materiales utilizados en el prototipo final, sin contar los costos de iteración, horas hombre, equipamiento técnico, ni el valor del software de monitoreo y control diseñado. Es decir, sólo se considerará los costos tangibles para poder replicar este prototipo funcional.

A continuación, se muestra en la tabla 1 el desglose de los costos del prototipo, presentando los valores en dólares y en pesos chilenos. Los artículos que presentan sus valores en dólares fueron importados directamente de china y se realizó la conversión con el valor del dólar del día 17 de octubre de 2016 el cual fue de USD\$670 (SII, 2016). Como resultado tenemos el costo total del prototipo funcional de **CLP\$296.804.**

Tabla 1 Análisis de Inversiones y Costos

N°	Artículo	Precio Unidad (USD)	Precio Unidad (CLP)	Cantidad	Subtotal	Porcentaje de incidencia
1	Válvula Solenoide	\$3,95	\$2.647	4	\$10.586	3,6%
2	Bomba Sanitaria	\$38,00	\$25.460	2	\$50.920	17,2%
3	Bomba Agua	\$14,99	\$10.043	1	\$10.043	3,4%
4	Calentador	\$14,83	\$9.936	2	\$19.872	6,7%
5	Fluxómetro	\$3,80	\$2.546	4	\$10.184	3,4%
6	Sensor Nivel	\$2,34	\$1.568	4	\$6.271	2,1%
7	Sensor Temperatura	\$1,14	\$764	3	\$2.291	0,8%
8	Peltier	\$1,78	\$1.193	1	\$1.193	0,4%
9	Sensor Peso	\$3,15	\$2.111	4	\$8.442	2,8%
10	Convertor Sensor Peso	\$0,58	\$389	1	\$389	0,1%
11	SSR	\$3,74	\$2.506	2	\$5.012	1,7%
12	Sensor Alcohol	\$1,52	\$1.018	1	\$1.018	0,3%
13	Arduino Mega	\$6,15	\$4.121	2	\$8.241	2,8%
14	RTC	\$1,20	\$804	1	\$804	0,3%
15	Modulo SD	\$0,80	\$536	2	\$1.072	0,4%
16	HMI	\$23,66	\$15.852	1	\$15.852	5,3%
17	Buzzer	\$0,13	\$87	1	\$87	0,0%
18	Tip 122	-	\$312	7	\$2.184	0,7%
27	Fuente de Poder	-	\$8.990	2	\$17.980	6,1%
28	Cables	-	\$8.390	1	\$8.390	2,8%
19	Disipador Aluminio	\$11,00	\$7.370	2	\$14.740	5,0%
20	Serpentín Cobre	-	\$7.350	1	\$7.350	2,5%
21	Madera Fermentador	-	\$20.820	1	\$20.820	7,0%
22	Plumavit Fermentador	-	\$10.680	1	\$10.680	3,6%
24	Manguera Alimentaria	-	\$2.700	4	\$10.800	3,6%
25	Recipiente Aluminio	-	\$17.900	2	\$35.800	12,1%
26	Cañerías y Fittings	-	\$15.782	1	\$15.782	5,3%
<b>Total</b>					<b>\$296.804</b>	<b>100,0%</b>

Fuente: Propia (2016)

Este desglose incluye todos los materiales, sensores, actuadores, cables que se adquirieron, exceptuando algunos artículos que ya se poseían o reciclaron de otras partes, como lo son los fierros utilizados para armar la estructura principal, el material aislante aluminizado del fermentador, el alimentador de lúpulos, entre otras cosas.

Se aprecia que la distribución de costos es relativamente equitativa, exceptuando los artículos de la fila 2 y 25 los cuales, entre los dos, representan un 29,3% del costo total del proyecto.

SOLO USO ACADÉMICO

## CAPITULO VI

### CONCLUSIONES

#### 6.1. ANALISIS DEL DISEÑO IMPLEMENTADO

Se diseña un prototipo de un sistema de elaboración de cerveza artesanal que permite controlar y/o monitorear las variables fundamentales a lo largo de todo el proceso como lo son: temperatura, tiempo, cantidad de líquido y producción de etanol.

El sistema almacena todos los datos de las variables fundamentales con el fin de recopilar información vital para su posterior análisis.

Autonomía parcial del proceso: se puede desligar lo máximo posible al usuario final, a excepción de algunas etapas, para las cuales es necesaria la intervención humana.

Precisión en el control de la temperatura, permitiendo alcanzar errores menores al 1% en su estado estacionario.

Se diseñan e implementan sistemas de seguridad en diferentes etapas de la elaboración, permitiendo generar al usuario final mensajes de alertas de un mal funcionamiento del sistema.

Se desarrolla una interfaz gráfica, amigable y fácil de utilizar. Se utilizó la pantalla táctil *NX4832T035* y su software editor, el cual muestra los parámetros e ilustraciones necesarias en todo el proceso de elaboración.

#### 6.2. DIFICULTADES ENCONTRADAS

A lo largo de este proyecto, se presentaron diversos inconvenientes y dificultades en las diferentes áreas del diseño e implementación, de las cuales, el área más problemática fue la plomería. Debido a que hubo diversas conexiones con pérdidas, mangueras que se desconectaban a mitad de proceso, etc., ocasionando que se iterara el diseño y fabricación.

#### 6.3. TIEMPO DEL DESARROLLO DEL PROTOTIPO

Al momento de desarrollar un prototipo se deben considerar diversos factores que afectan el tiempo de fabricación como lo son las diferentes iteraciones de diseño,

plazos de importación de cada elemento (sensor, actuador, entre otros), fabricación, ensamblaje, pruebas, entre otras cosas. Considerando lo anterior, el tiempo de fabricación de este prototipo se realizó dentro de un año.

No obstante, el tiempo de re-implementación de este prototipo es considerablemente menor, ya que, debido a la investigación y desarrollo realizado, los materiales y los métodos de ensamblaje ya se encuentran identificados.

#### **6.4. RESULTADOS Y TRABAJOS FUTUROS**

Utilizando sensores de bajo costo y generando una buena calibración, se puede alcanzar excelentes resultados, como los vistos con los parámetros del controlador PI.

Utilizando la información de las variables fundamentales, almacenadas en la tarjeta de memoria SD, se pueden revisar datos del sistema con el fin de mejorar la calidad del producto final.

Este proyecto brinda la posibilidad de acercar a más personas al arte de la elaboración de cerveza artesanal, ya que facilita su elaboración y se aprende de los diferentes factores que afectan el producto final, mediante el análisis de los datos guardados.

La implementación de este sistema corresponde al primer prototipo de un proyecto más ambicioso, el cual posee diversas proyecciones a futuro, como los mencionados a continuación:

- Lograr la autonomía completa del sistema, permitiendo la deligación total del maestro cervecero a lo largo de todo el proceso de elaboración de cerveza artesanal.
- Completa flexibilidad y adaptabilidad tanto en el hardware como en el software con el fin de poder adaptarse completamente a las necesidades de cada usuario final.
- Comprobar la repetitividad del producto final, mediante estudios químicos y catadores certificados, logrando así un sistema capaz de reiterar el producto las veces que se deseen.

- Generar una red social en la cual se compartan experiencias, metodologías, errores de elaboración, mejoras, recetas, entre otras características, permitiendo abrir las puertas de una comunidad solidaria a futuros emprendedores en el ámbito de la elaboración de cerveza artesanal.

SOLO USO ACADÉMICO

## BIBLIOGRAFIA

- Atkins, P., Jones, L., & Gismondi, M. (2006). *Principios de química* (1st ed.). Buenos Aires: Médica Panamericana.
- Beauregard, B. (2011). *Arduino*. Obtenido de <http://playground.arduino.cc/Code/PIDLibrary>
- Boss, B. (2016). *Brew Boss*. Obtenido de <http://www.brew-boss.com/>
- Denia, C. Castromil, J. (2015). *20 minutos*. Obtenido de <http://blogs.20minutos.es/la-gulateca/2015/02/19/brewie-la-maquina-para-hacer-cerveza-en-casa-de-forma-automatica/>
- DMedicina. (2016). *DMedicina*. Obtenido de <http://www.dmedicina.com/vida-sana/alimentacion/diccionario-de-alimentacion/glucosa.html>
- García, V. B. (2013). *Ohmios*. Obtenido de <http://ohmios.es/2013/05/01/efecto-peltier/>
- Grainfather. (2016). *Grainfather*. Obtenido de <http://www.grainfather.com/shop/grainfather/grainfather.html>
- Hernández, A. (2003). *Microbiología Industrial*. Costa Rica: EUNED.
- Higieneysan. (2016). *Higieneysan*. Obtenido de <http://www.higieneysan.cl/sanitizacion-desinfeccion-eliminar-microorganismos-bacterias-hongos-levaduras-virus-higiene-ambiental.htm>
- Indiegogo. (2015). *Indiegogo*. Obtenido de <https://www.indiegogo.com/projects/brewie-world-s-first-fully-automated-home-brewery-beer-brewery#/>
- Ironside, D. (2015). *Idiot's Guides: Homebrewing*. Indianapolis, Estados Unidos: Alpha.
- Mathews, C., Van Holde, K., Ahern, K., & González de Buitrago, J. (2002). *Bioquímica* (1st ed.). Madrid: Addison Wesley.
- Mazzone, V. (2002). *Controladores PID*. Universidad de Quilmes. Obtenido de <http://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>
- Morales, F. J. (2001). *TERMODINAMICA: UNA GUIA DE CLASE*. UNIVERSIDAD DE SEVILLA. SECRETARIADO DE PUBLICACIONES. Obtenido de <http://www.casadellibro.com/libro-termodinamica-una-guia-de-clase/9788447206995/857160>
- Netto, R. S. (2016). *Fisicanet*. Obtenido de [http://www.fisicanet.com.ar/fisica/termodinamica/tb03\\_conductividad.php](http://www.fisicanet.com.ar/fisica/termodinamica/tb03_conductividad.php)
- Profesorenlinea. (2015). Obtenido de <http://www.profesorenlinea.cl/Ciencias/Glucosa.html>
- Ragoni, M. (2016). *Ceresvis*. Obtenido de <http://www.ceresvis.com/index.php/notas-cerveceras/elaboracion-de-cerveza/avanzados/item/4-quimica-del-lupulo-primera-parte>
- SII. (2016). *SII*. Obtenido de <http://www.sii.cl/pagina/valores/dolar/dolar2016.htm>
- Turevskiy, A. (2012). *Mathworks*. Obtenido de <https://www.mathworks.com/discovery/pid-tuning.html>

## ANEXOS

### ANEXO N°1. Código de las diferentes funciones.

#### 1. Código Parámetros

```
while(Estado_Parametro==0)
{
    Datos_Cerveza();
}
while(Estado_Boton==0){
    presionar_boton();
}
scale.tare(); //Reset the scale to 0
//Mensaje de colocar la malta
Estado_Boton=0;
while (Estado_Boton==0)
{
    Malta=Pesar()*1000;//se pasan a gramos
    if (Malta>0)
    {
        malt= String(Malta); //Guardar en SD
        N_Var_Malta.setValue(Malta);//Se envia el peso de la malta al nextion
    }else{
        N_Var_Malta.setValue(0);//Se envia el peso de la malta al nextion
    }
    Serial.print("Malta Pesa= ");
    Serial.print(Malta);
    Serial.println(" g");
    Serial.print("Estado boton= ");
    Serial.println(Estado_Boton);
    presionar_boton();
}
Cantidad_Agua_Inicial=(Malta*Parametros[6]); // Se multiplica por la relacion
malta/Agua elegida
//Guardar datos en SD
printDate();
previousMillis=0;
interval=1;
imprimir="Temp Macerado;Tiempo Macerado;Temp Hervido;Tiempo Hervido;Tiempo Lupulo
1;Tiempo Lupulo 2;Tiempo Lupulo 3;Cant Malta;Relacion Malt/H2O;Cantidad de
Mosto;Hora;Fecha";
sd(interval,imprimir);
imprimir=String(Parametros[0])+";" +String(Parametros[1])+";" +100+";" +String(Parametros[
2])+";" +String(Parametros[3])+";" +String(Parametros[4])+";" +String(Parametros[5])+";" +m
alt+";" +String(Parametros[6])+";" +String(Cantidad_Agua_Inicial)+";" +mensaje+";" +msjfull
;
sd(interval,imprimir);
imprimir="Setpoint 1;Temperatura 1;Output 1;Setpoint 2;Temperatura 2;Output 2;Tiempo
Faltante;Tiempo Proceso;Hora;Peso Mosto;Cantidad Agua;Etapa;Alerta";
sd(interval,imprimir);
```

## 2. Código Calentando

```
Que_Proceso=0;//Proceso de Calentado
HMI(Que_Proceso);
void Calentando()
{
    Etapa="Calentando";
    Direccion_PID=0;//Directamente proporcional (0= Direct ; 1=Reverse)
    Setpoint=Parametros[0]; //Temp de maceracion
    Que_Flujometro=0;
    Que_PID=TRUE; //PID Fondo 1

    digitalWrite(SSR[1],LOW);
    digitalWrite(Solenoides[0],HIGH);
    delay(4000);
    digitalWrite(Bomba[0],HIGH);
    while (Proceso_Calentado==0 && Proceso_Maceracion==0 && Proceso_Hervido==0)
    {
        Proceso(Etapa,Parametros[1]);
    }
    digitalWrite(Solenoides[0],LOW);
    digitalWrite(Bomba[0],LOW);
    analogWrite(SSR[0],0);
}

//PAGINA DE MENSAJES
N_Pag_Mensajes.show();
nex.refresh();
nex.sendCommand("vis t0,1");
nex.sendCommand("vis t1,1");
nex.sendCommand("vis t2,1");
N_Str_Mensaje0.setText("Traspasar ");
N_Str_Mensaje1.setTextAsNumber(Cantidad_Agua_Inicial);
N_Str_Mensaje2.setText(" ml al Fondo 2");
//Esperar que se haya presionado el boton
Estado_Boton=0;
while(Estado_Boton==0){
    presionar_boton();
}

printDate();
//MOSTRAR EN DISPLAY
//FONDO1, FONDO 2,TRASP, TEMP1,TEMP2, HORA
N_Pag_Pag1.show();
nex.refresh();
nex.sendCommand("vis p1,1");//Fondo 1
nex.sendCommand("vis p3,1");//Fondo 2
nex.sendCommand("vis p6,1");//Traspaso
nex.sendCommand("vis p8,1");//Estado SD
nex.sendCommand("vis p9,1");//Foto Hora
nex.sendCommand("vis n3,1");//Flujometro
nex.sendCommand("vis n4,1");//Hora
nex.sendCommand("vis n5,1");//Minuto
nex.sendCommand("vis t0,1");//Titulo Etapa
N_Str_Etapa.setText("Traspaso");//Se envia la etapa
```

```

//SE INGRESA LA CANTIDAD DE AGUA CALCULADA AL FONDO 2
Etapa="AguaInicial";
Que_Flujometro=1;
digitalWrite(Solenoides[1],HIGH);
digitalWrite(Solenoides[0],LOW);
digitalWrite(Bomba[0],HIGH);
boolean valor;
valor=digitalRead(S_Nivel[2]);
while (valor==FALSE)
{
    valor=digitalRead(S_Nivel[2]);
}
scale.tare(); //Reset the scale to 0
//Cuando se tenga el sensor de nivel, poner la condicion que cuando se active comienze
a contar los ml
Cant_Agua(Cantidad_Agua_Inicial,Que_Flujometro);
h2o = String(totalMilliLitres); //Guardar en SD
totalMilliLitres=0; //reinicio de la variable
digitalWrite(Solenoides[1],LOW);
digitalWrite(Bomba[0],LOW);
delay(3000); //Para que caiga toda el agua de la manguera

Serial.print("El Agua antes de Macerado Pesa= ");
Absorcion[0]=Pesar()*1000; //Es el peso del agua antes del macerado
wort=String(Absorcion[0]);
Serial.print(Absorcion[0]);
Serial.println(" grs");
//Guardar en SD
interval=10;
imprimir=set1 +";"+ temp1 +";"+out1+";"+set2 +";"+ temp2
+";"+out2+";"+Tiempo_Faltante+";"+Tiempo+";"+mensaje+";"+wort+";"+h2o+";"+Etapa+";"+ale
rt; //SD
sd(interval,imprimir);

```

### 3. Código Macerado

```
Que_Proceso=1;
HMI(Que_Proceso);
void Macerado()
{
    Etapa="Macerado";
    Direccion_PID=0;//Directamente proporcional (0= Direct ; 1=Reverse)
    Setpoint=Parametros[0]; //Temp de maceracion
    Que_Flujometro=2;
    Que_PID=FALSE;//Fondo 2

    digitalWrite(Solenoides[2],HIGH);
    delay(4000);
    digitalWrite(Bomba[1],HIGH);
    delay(20000);//para que el sensor de temp se estabilize y asi no dispara los
valores del PID
    while (Proceso_Calentado==1 && Proceso_Maceracion==0 && Proceso_Hervido==0)
    {
        Proceso(Etapa,Parametros[1]);
    }
    digitalWrite(Solenoides[0],LOW); //Apaga la bomba y solenoide del fondo 1
    digitalWrite(Bomba[0],LOW);
    digitalWrite(Solenoides[2],LOW);
    digitalWrite(Bomba[1],LOW);
    digitalWrite(Bomba[1],LOW);
    digitalWrite(SSR[0],LOW);
    digitalWrite(SSR[1],LOW);
}

N_Pag_Mensajes.show();
nex.refresh();
nex.sendCommand("vis t0,1");
nex.sendCommand("vis t1,1");
nex.sendCommand("vis t2,1");
N_Str_Mensaje0.setText("Levante la malta del recipiente...");
Estado_Boton=0;
while(Estado_Boton==0){
    presionar_boton();
}

Absorcion[1]=Pesar()*1000;
wort=String(Absorcion[1]);
Etapa="Termino Macerado";
//Guardar en SD
interval=10;
imprimir=set1 +" "+" temp1 +" "+out1+" "+set2 +" "+" temp2
"+" "+out2+" "+Tiempo_Faltante+" "+Tiempo+" "+mensaje+" "+wort+" "+h2o+" "+Etapa+" "+ale
rt;//SD
sd(interval,imprimir);

//Agua Lavado= Agua Absorcion + Agua Evaporada;
Cant_Agua_Lavado= (Absorcion[0]-Absorcion[1])+Evaporado;//

//LAVADO O SPARGING
```

```
N_Pag_Mensajes.show();
nex.refresh();
nex.sendCommand("vis t0,1");
nex.sendCommand("vis t1,1");
nex.sendCommand("vis t2,0");
nex.sendCommand("vis t3,1");
nex.sendCommand("vis n0,1");
N_Str_Mensaje0.setText("Coloque la nalta en el recipiente para comenzar el Lavado...");
N_Str_Mensaje1.setText("Con= ");
N_Var_Var.setValue(Cant_Agua_Lavado);
N_Str_Unidad.setText(" mL");
Estado_Boton=0;
while(Estado_Boton==0){
    presionar_boton();
}
```

SOLO USO ACADÉMICO

#### 4. Código Lavado

```
Que_Proceso=2;
HMI(Que_Proceso);
void Lavado()
{
    Etapa="Lavado";

    Que_Flujometro=1;

    digitalWrite(Solenoides[1],HIGH);
    digitalWrite(Bomba[0],HIGH);
    Cant_Agua(Cant_Agua_Lavado,Que_Flujometro);
    h2o = String(totalMilliLitres); //Guardar en SD
    totalMilliLitres=0;//Reinicio de la variable
    digitalWrite(Solenoides[1],LOW);
    digitalWrite(Bomba[0],LOW);
}

N_Pag_Mensajes.show();
nex.refresh();
nex.sendCommand("vis t0,1");
N_Str_Mensaje0.setText("RETIRAR LA MALTA !!!!");
Estado_Boton=0;
while(Estado_Boton==0){
    presionar_boton();
}
//Mosto=(Pesar()*1000)-Malta;
Mosto=Pesar()*1000;
wort=String(Mosto);

//Guardar en SD
interval=10;
imprimir=set1 +";"+ temp1 +";"+out1+";"+set2 +";"+ temp2
+";"+out2+";"+Tiempo_Faltante+";"+Tiempo+";"+mensaje+";"+wort+";"+h2o+";"+Etapa+";"+ale
rt;//SD
sd(interval,imprimir);

nex.refresh();
nex.sendCommand("vis t0,1");
nex.sendCommand("vis t1,0");
nex.sendCommand("vis t3,0");
nex.sendCommand("vis n0,0");
N_Str_Mensaje0.setText("Coloque Serpentin y Alimentador de Lupulos...");
Estado_Boton=0;
while(Estado_Boton==0){
    presionar_boton();
}
```

## 5. Código Hervido

```
Que_Proceso=3;
HMI(Que_Proceso);
void Hervido()
{
    Etapa="Coccion";

    Direccion_PID=0; //Directamente proporcional (0= Direct ; 1=Reverse)
    Setpoint=temp_hervido; //Temperatura del hervido
    Que_Flujometro=2;
    Que_PID=FALSE; //Fondo 2
    PIDFondo2.SetTunings(aggKp, aggKi, aggKd);
    //ACTIVAR SOLENOIDES Y BOMBAS
    digitalWrite(Solenoides[2],HIGH);
    digitalWrite(Bomba[1],HIGH);
    delay(4000);
    while (Proceso_Calentado==1 && Proceso_Maceracion==1 && Proceso_Hervido==0)
    {
        Proceso(Etapa,Parametros[2]);
    }
    digitalWrite(Solenoides[2],LOW);
    digitalWrite(Bomba[1],LOW);
    digitalWrite(SSR[1],LOW);
}
```

SOLO USO ACADÉMICO

## 6. Código Enfriado

```
Que_Proceso=4;
HMI(Que_Proceso);

void Enfriado()
{
    int contador=0;
    Etapa="Enfriado";
    Que_Flujometro=2;

    digitalWrite(Solenoides[2],HIGH);
    digitalWrite(Bomba[1],HIGH); //Bomba recirculacion macerado
    digitalWrite(Bomba[2],HIGH); // Bomba del serpiente
    delay(4000);

    while(contador<80)
    {
        Alerta(Que_Proceso);
        sensors.requestTemperatures(); //Prepara el sensor de temp para la
lectura
        Input= sensors.getTempC(TempSensor_Macerado);
        temp2=String(Input);
        set2=String(temp_enfriado);
        printDate();
        N_Var_Temp2.setValue(Input);
        nex.sendCommand("p5.pic=12");
        //Guardar en SD
        interval=10000;
        imprimir=set1 +" "+ temp1 +" "+out1+" "+set2 +" "+ temp2
+" "+out2+" "+Tiempo_Faltante+" "+Tiempo+" "+mensaje+" "+wort+" "+h2o+" "+Etapa+" "+ale
rt;//SD
        sd(interval,imprimir);

        if(Input==Input_anterior)
        {
            Input_anterior=Input;
            cont_no++;
        }else{cont_no=0;}

        if (cont_no>=80)
        {
            digitalWrite(Solenoides[2],LOW);
            digitalWrite(Bomba[1],LOW); //Bomba recirculacion macerado
            digitalWrite(Bomba[2],LOW); // Bomba del serpiente
            nex.refresh();
            nex.sendCommand("vis t0,1");
            nex.sendCommand("vis t1,0");
            nex.sendCommand("vis t3,0");
            nex.sendCommand("vis n0,0");
            N_Str_Mensaje0.setText("Cambiar Agua del enfriador");
            Estado_Boton=0;
            while(Estado_Boton==0){
                presionar_boton();
            }
        }
    }
}
```

```
        if(Input<=temp_enfriado)
        {
            contador++;
        }else{contador=0;}
    }
    if (contador>=80)
    {
        step(false); //Gira el motor para botar la levadura
        digitalWrite(Solenoides[2],LOW);
        digitalWrite(Bomba[1],LOW); //Bomba recirculacion macerado
        digitalWrite(Bomba[2],LOW); // Bomba del serpentín
        contador=0;
        Proceso_Enfriado=1;
    }
}
```

SOLO USO ACADÉMICO

## 7. Código Traspaso Fermentador

```
N_Pag_Mensajes.show();
nex.refresh();
nex.sendCommand("vis t0,1");
N_Str_Mensaje0.setText("COLOCAR RECIPIENTE FERMENTADOR");
Estado_Boton=0;
while(Estado_Boton==0){
    presionar_boton();
}

Que_Proceso=5;
HMI(Que_Proceso);
Traspaso();
N_Var_Cerveza.setValue(totalMilliLitres);
Estado_Boton=0;
while(Estado_Boton==0){
    presionar_boton();
}
totalMilliLitres=0;
N_Pag_Mensajes.show();
nex.refresh();
nex.sendCommand("vis t0,1");
N_Str_Mensaje0.setText("TERMINO ELABORACION DE CERVEZA");
Estado_Boton=0;
while(Estado_Boton==0){
    presionar_boton();
}
```

## 8. Código Fermentador

```
unsigned long currentMillis = millis();
Direccion_PID=0;//Directamente proporcional (0= Direct ; 1=Reverse)
Setpoint=23;
set=String(Setpoint);
unsigned long currentMillis = millis();

Uso_PID(Direccion_PID,Setpoint);
if (currentMillis - previousMillis >= interval)
{
    Serial.print("Contador= ");
    Serial.println(contador);
    datos();
    previousMillis = currentMillis;
}
```

## 9. Código Proceso

```
void Proceso (String Etapa,int Tiempo)
{
  int rango;//Rango de error de temperatura

  Uso_PID(Direccion_PID,Setpoint);

  printDate();
  Alerta(Que_Proceso);
  //Datos q se deben guardar en la SD
  interval=10000;
  sd(interval,imprimir);

  if (abs(Input-temp_anterior)<=0.06)//no ha cambiado mucho de temperatura
  {
    cont_tiempo++;
  if (cont_tiempo==20)
  {
    cont_tiempo=0;
    //agregar parametros agresivos al pid
    PIDFondo2.SetTunings(aggKp, aggKi, aggKd);
    PIDFondo1.SetTunings(aggKp, aggKi, aggKd);
  }
  }else{cont_tiempo=0;}
  temp_anterior=Input;

  gap = abs(Setpoint-Input);
  if (Que_Proceso==3)//Hervido
  {
    rango=13;//-+13°C de error, ya que el agua no hierve a los 100°C
  }else{rango=1;}
  if(gap<=rango)
  {
    cont++;
  if (cont==30)//Quiere decir que esta estable la temperatura
  {
    cont=0;
    if (Etapa=="Calentando")
    {
      Proceso_Calentado=1;
      return; //Vuele al loop
    }
    if (Etapa=="Macerado")
    {
      //tiempo de termino de macerado
      Tiempo=Parametros[1];
      Serial.print("Tiempo Macerado= ");
      Serial.println(Tiempo);
    }
    if (Etapa=="Coccion")
    {
      //tiempo de termino de hervido
      Tiempo=Parametros[2];
      Serial.print("Tiempo Hervido= ");
      Serial.println(Tiempo);
      Minuto_Lupulo[0]=Parametros[3];
    }
  }
}
```

```
        Minuto_Lupulo[1]=Parametros[4];
        Minuto_Lupulo[2]=Parametros[5];
    }
    unsigned long Cronometro=millis();//comienza el cronometro
    Cronometro_Anterior=Cronometro;
    cont_lupulo=0; //Comienza el contador de lupulos
    //Aqui se queda procesando hasta q el macerado o coccion terminen
    Temp_Estable();
}
}else{cont=0;}
}
```

SOLO USO ACADÉMICO

## 10. Código Temp Estable

```
void Temp_Estable()
{
    while(Cronometro_minutos<Tiempo)
    {
        Alerta(Que_Proceso);
        Cronometro=millis();
        Cronometro_minutos=(Cronometro-Cronometro_Anterior)/60000;
        Tiempo_Faltante= Tiempo-Cronometro_minutos;

        Que_PID=FALSE; //Fondo 2
        Uso_PID(Direccion_PID,Setpoint);
        if (Etapa=="Macerado")
        {
            nex.sendCommand("vis p0,1");//Recirculado Fondo 1
            nex.sendCommand("vis p1,1");//Fondo 1
            nex.sendCommand("vis p4,1");//Calentador 1
            nex.sendCommand("vis n0,1");//Temp Fondo 1
            nex.sendCommand("vis p7,1");//para ver el reloj de arena
            nex.sendCommand("vis n2,1");//para ver cuanto le falta
            N_Var_Tiempo.setValue(Tiempo_Faltante);//Se envia el tiempo

            faltante

            N_Var_Hora.setValue(hour);
            N_Var_Minuto.setValue(minute);
            Que_PID=TRUE; //Fondo 1
            digitalWrite(Solenoides[0],HIGH);
            digitalWrite(Bomba[0],HIGH);
            Uso_PID(Direccion_PID,Parametros[0]);
            Que_Proceso=0;
            Alerta(Que_Proceso);
            Que_Proceso=1;
        }

        //ACCION AL COMPLETAR LA HORA DE CADA LUPULO
        //ACTIVAR HOPS FEEDER
        if (Etapa=="Coccion" ||Etapa== "Lupulo")
        {
            nex.sendCommand("vis p3,1");//para ver el reloj de arena
            nex.sendCommand("vis n4,1");//para ver cuanto le falta
            nex.sendCommand("vis p9,1");//Reloj
            nex.sendCommand("vis n5,1");//Hora
            nex.sendCommand("vis n6,1");//Minuto
            nex.sendCommand("vis p7,1");

            N_Var_Tiempo2.setValue(Tiempo_Faltante);//Se envia el tiempo

            faltante

            N_Var_Hora2.setValue(hour);
            N_Var_Minuto2.setValue(minute);
            for (i=0;i<3;i++)
            {
                if (Tiempo_Faltante==Minuto_Lupulo[i])
                {
                    Serial.print("Se ha ingresado el Lupulo ");
                    Serial.println(cont_lupulo+1);
                }
            }
        }
    }
}
```



```

        Mosto=Pesar()*1000;
        wort=String(Mosto);
        printDate();
        // Guarda datos en sd
        interval=10000;
        mostrar();//imprime en pantalla los datos
    }
    //Aqui termina el macerado o coccion
    if (Cronometro_minutos==Tiempo)
    {
        Cronometro=0;//se reinicia el cronometro
        Cronometro_Anterior=0;
        Cronometro_minutos=0;

        if (Etapa== "Macerado")
        {
            Proceso_Maceracion=1; // Significa que el proceso de maceracion ha
            sido cumplido
        }
        if (Etapa== "Coccion")
        {
            Proceso_Hervido=1;
        }
    }
}

```

SOLO USO ACADÉMICO

## 11. Código PID

```
void PID(int Estado,double Setpoint)
{
    if (Que_PID==TRUE) //PID Fondo 1
    {
        PIDFondo1.SetControllerDirection(Estado); //Cambia el estado directa o
reversa
        set1 = String(Setpoint);
        sensors.requestTemperatures(); //Prepara el sensor de temp para la
lectura
        Input= sensors.getTempC(TempSensor_Agua);
        N_Var_Temp1.setValue(Input);

        PIDFondo1.Compute();
        analogWrite(SSR[0],Output1);
        out1=String(Output1);//Guardar en SD
        temp1=String(Input); //Temperatura del sensor
        Output1=0;
    }
    if (Que_PID==FALSE) //PID Fondo 2
    {
        PIDFondo2.SetControllerDirection(Estado); //Cambia el estado directa o
reversa
        set2 = String(Setpoint);
        sensors.requestTemperatures(); //Prepara el sensor de temp para la
lectura
        Input= sensors.getTempC(TempSensor_Macerado);
        if (Etapa=="Macerado")
        {
            N_Var_Temp2.setValue(Input);
        }else{
            N_Var_Temp.setValue(Input);//Hervido
        }

        PIDFondo2.Compute();
        analogWrite(SSR[1],Output2);
        out2=String(Output2);//Guardar en SD
        temp2=String(Input); //Temperatura del sensor
        Output2=0;
    }
}
```

## 12. Código Datos

```
void datos()
{
    revisar();//si son invalidos dar alerta
    guardar();
    if (dato < 1440) //1 dia
    {
        Dia=Dia+ABV; //suma de todos
        dato++;
        return;
    }else
    {
        //paso un dia
        promedio=Dia/dato; //Se obtiene el promedio de todos los valores del dia
        Dias[cont_dia]=promedio;

        if(cont_dia!=0)
        {
            condicion=Dias[cont_dia]/Dias[cont_dia-1];
            condicion=abs(condicion-1);
            if(condicion<0.04)
            {
                cont_termino++;
            }else{cont_termino=0;}
        }
        dato=0;
        cont_dia++;
    }
    if(cont_termino==3)//tres dias seguidos con <4% de variacion
    {
        while(1)
        {
            //Tono de alerta
            tone(7,1000);
            delay(100);
            noTone(7);
            delay(100);
        }
    }
}
```

### 13. Código Alerta

```
void Alerta(int Proceso)
{
    char *mensaje_alerta_flujometro="";
    char *mensaje_alerta="";
    int estado[2];
    boolean FlujoMetro_Malo=FALSE;
    boolean Nivel_Malo=TRUE;
    switch (Proceso)
    {
        case 0://Calentando
            //Revisar FlujoMetro
            pulseCount=0;
            attachInterrupt(digitalPinToInterrupt(flujoMetro[Que_FlujoMetro]),
pulseCounter, RISING);
            delay(500);
            detachInterrupt(flujoMetro[Que_FlujoMetro]);
            if (pulseCount==0)
            {
                FlujoMetro_Malo=TRUE;
                mensaje_alerta_flujometro="Error Flow/Bomba/Sol F.1";
                alert="F1-F0";
            }else{FlujoMetro_Malo=FALSE;alert="OK";}

            //Revisar Sensor Nivel
            estado[0]=digitalRead(S_Nivel[0]);
            estado[1]=digitalRead(S_Nivel[1]);
            if(estado[0]==1 || estado[1]==1)
            {
                delay(300);
                estado[0]=digitalRead(S_Nivel[0]);
                estado[1]=digitalRead(S_Nivel[1]);
            }

            if (estado[0]==0 && estado[1]==0)//ALERTA
            {
                mensaje_alerta="AGREGUE AGUA AL FONDO 1";
                alert="F1-SN.Inf";
                Nivel_Malo=TRUE;
            }

            if (estado[0]==0 && estado[1]==1)//CASO IMPOSIBLE
            {
                mensaje_alerta="ARREGLAR SENSOR SUPERIOR FONDO 1";
                alert="F1-SN.Sup";
                Nivel_Malo=TRUE;
            }

            if (estado[0]==1 && estado[1]==0)//OK
            {
                mensaje_alerta="Niveles OK";
                if (alert!="F1-F0")
                {
                    alert="OK";
                }
                Nivel_Malo=FALSE;
            }

            if (estado[0]==1 && estado[1]==1)//ALERTA
```

el proceso

```
{
    mensaje_alerta="FONDO 1 SE REBALSA";
    alert="F1-SN.Sup";
    Nivel_Malo=TRUE;
}
//Accion si ocurre error
if (Nivel_Malo==TRUE || Flujometro_Malo==TRUE)//Hay algo malo en
{
    digitalWrite(Bomba[0],LOW);
    digitalWrite(Solenoides[0],LOW);
    digitalWrite(SSR[0],LOW);
    digitalWrite(Bomba[1],LOW);
    digitalWrite(Solenoides[2],LOW);
    digitalWrite(SSR[1],LOW);
    N_Pag_Mensajes.show();
    nex.refresh();
    nex.sendCommand("vis t2,1");
    nex.sendCommand("t2.font=0");
    N_Str_Mensaje2.setText(mensaje_alerta);
    nex.sendCommand("t0.font=0");
    nex.sendCommand("vis t1,1");
    N_Str_Mensaje0.setText(mensaje_alerta_flujometro);
    Estado_Boton=0;
    while(Estado_Boton==0){
        presionar_boton();
    }
    if (Etapa=="Macerado")
    {
        HMI(1);
        digitalWrite(Solenoides[0],HIGH);
        digitalWrite(Bomba[0],HIGH);
        digitalWrite(Solenoides[2],HIGH);
        digitalWrite(Bomba[1],HIGH);
        return;
    }
    HMI(Proceso);
    digitalWrite(Solenoides[0],HIGH);
    digitalWrite(Bomba[0],HIGH);
}
break;
case 1://Macerado
//Revisar Flujometro
pulseCount=0;
attachInterrupt(digitalPinToInterrupt(flujometro[Que_Flujometro]),
pulseCounter, RISING);
delay(500);
detachInterrupt(flujometro[Que_Flujometro]);
if (pulseCount==0)
{
    Flujometro_Malo=TRUE;
    mensaje_alerta_flujometro="Error Flow/Bomba/Sole F.2";
    alert="F2-F2";
}else{Flujometro_Malo=FALSE;alert="OK";}
//Revisar Sensor Nivel
estado[0]=digitalRead(S_Nivel[2]);
estado[1]=digitalRead(S_Nivel[3]);
if(estado[0]==1 ||estado[1]==1)
```

```

{
    delay(300);
    estado[0]=digitalRead(S_Nivel[2]);
    estado[1]=digitalRead(S_Nivel[3]);
}
if (estado[0]==0 && estado[1]==0)//ALERTA
{
    //NO HAY AGUA EN EL FONDO 2
    mensaje_alerta="NO SE TRASPASO AGUA AL FONDO 2!!!!";
    alert="F2-SN.Inf";
    Nivel_Malo=TRUE;
}
if (estado[0]==0 && estado[1]==1)//CASO IMPOSIBLE
{
    mensaje_alerta="ARREGLAR SENSOR SUPERIOR FONDO 2";
    alert="F2-SN.Sup";
    Nivel_Malo=TRUE;
}
if (estado[0]==1 && estado[1]==0)//OK
{
    mensaje_alerta="Niveles OK";
    if (alert!="F2-F2")
    {
        alert="OK";
    }
    Nivel_Malo=FALSE;
}
if (estado[0]==1 && estado[1]==1)//ALERTA
{
    mensaje_alerta="FONDO 2 SE REBALSA!!!!!!";
    alert="F2-SN.Sup";
    Nivel_Malo=TRUE;
}
//Accion si ocurre error
if (Nivel_Malo==TRUE || Flujometro_Malo==TRUE)//Hay algo malo en
el proceso
{
    digitalWrite(Bomba[0],LOW);
    digitalWrite(Solenoides[0],LOW);
    digitalWrite(SSR[0],LOW);
    digitalWrite(Bomba[1],LOW);
    digitalWrite(Solenoides[2],LOW);
    digitalWrite(SSR[1],LOW);
    N_Pag_Mensajes.show();
    nex.refresh();
    nex.sendCommand("vis t2,1");
    nex.sendCommand("t2.font=0");
    N_Str_Mensaje2.setText(mensaje_alerta);
    nex.sendCommand("t0.font=0");
    nex.sendCommand("vis t1,1");
    N_Str_Mensaje0.setText(mensaje_alerta_flujometro);
    Estado_Boton=0;
    while(Estado_Boton==0){
        presionar_boton();
    }
    HMI(Proceso);
    digitalWrite(Solenoides[2],HIGH);
}

```

```

        digitalWrite(Bomba[1],HIGH);
    }
break;
case 2://Lavado
//Revisar Flujo metro
if (pulseCount==0)
{
    Flujo metro_Malo=TRUE;
    mensaje_alerta_flujo metro="Error Flow/Bomba/Sole F.1";
    alert="F1-F1";
}else{Flujo metro_Malo=FALSE;alert="OK";}
//Revisar Sensor Nivel
estado[0]=digitalRead(S_Nivel[0]);
estado[1]=digitalRead(S_Nivel[1]);
if(estado[0]==1 ||estado[1]==1)
{
    delay(300);
    estado[0]=digitalRead(S_Nivel[0]);
    estado[1]=digitalRead(S_Nivel[1]);
}
if (estado[0]==0 && estado[1]==0)//ALERTA
{
    mensaje_alerta="NO HAY SUFICIENTE AGUA PARA EL LAVADO";
    alert="F1-SN.Inf";
    Nivel_Malo=TRUE;
}
if (estado[0]==0 && estado[1]==1)//CASO IMPOSIBLE
{
    mensaje_alerta="ARREGLAR SENSOR SUPERIOR FONDO 1";
    alert="F1-SN.Sup";
    Nivel_Malo=TRUE;
}
if (estado[0]==1 && estado[1]==0)//OK
{
    mensaje_alerta="Niveles OK";
    if (alert!="F1-F1")
    {
        alert="OK";
    }
    Nivel_Malo=FALSE;
}
if (estado[0]==1 && estado[1]==1)//ALERTA
{
    mensaje_alerta="FONDO 1 SE REBALSA";
    alert="F1-SN.Sup";
    Nivel_Malo=TRUE;
}
if (Nivel_Malo==TRUE || Flujo metro_Malo==TRUE)//Hay algo malo en
el proceso
{
    digitalWrite(Bomba[0],LOW);
    digitalWrite(Solenoides[1],LOW);
    digitalWrite(SSR[0],LOW);
    N_Pag_Mensajes.show();
    nex.refresh();
    nex.sendCommand("vis t2,1");
    nex.sendCommand("t2.font=0");
}

```

```

        N_Str_Mensaje2.setText(mensaje_alerta);
        nex.sendCommand("t0.font=0");
        nex.sendCommand("vis t1,1");
        N_Str_Mensaje0.setText(mensaje_alerta_flujometro);
        Estado_Boton=0;
        while(Estado_Boton==0){

                presionar_boton();

        }
        HMI(Proceso);
        digitalWrite(Solenoides[1],HIGH);
        digitalWrite(Bomba[0],HIGH);

    }
    //Accion si ocurre error
break;
case 3://Hervido
    //Revisar Flujo metro
    pulseCount=0;
    attachInterrupt(digitalPinToInterrupt(flujometro[Que_Flujo metro]),
pulseCounter, RISING);
    delay(500);
    detachInterrupt(flujometro[Que_Flujo metro]);
    if (pulseCount==0)
    {
        Flujo metro_Malo=TRUE;
        mensaje_alerta_flujometro="Error Flow/Bomba/Sol F.2";
        alert="F2-F2";
    }else{Flujo metro_Malo=FALSE;alert="OK";}
    //Revisar Sensor Nivel
    estado[0]=digitalRead(S_Nivel[2]);
    estado[1]=digitalRead(S_Nivel[3]);
    if(estado[0]==1 || estado[1]==1)
    {
        delay(300);
        estado[0]=digitalRead(S_Nivel[2]);
        estado[1]=digitalRead(S_Nivel[3]);
    }
    if (estado[0]==0 && estado[1]==0)//ALERTA
    {
        mensaje_alerta="NO HAY AGUA EN EL FONDO 2";
        alert="F2-SN.Inf";
        Nivel_Malo=TRUE;
    }
    if (estado[0]==0 && estado[1]==1)//CASO IMPOSIBLE
    {
        mensaje_alerta="ARREGLAR SENSOR SUPERIOR FONDO 2";
        alert="F2-SN.Sup";
        Nivel_Malo=TRUE;
    }
    if (estado[0]==1 && estado[1]==0)//OK
    {
        mensaje_alerta="Niveles OK";
        if (alert!="F2-F2")
        {
            alert="OK";
        }
        Nivel_Malo=FALSE;
    }

```

```

}
if (estado[0]==1 && estado[1]==1)//ALERTA
{
    mensaje_alerta="FONDO 2 SE REBALSA";
    alert="F2-SN.Sup";
    Nivel_Malo=TRUE;
}
//Accion si ocurre error
if (Nivel_Malo==TRUE || Flujometro_Malo==TRUE)//Hay algo malo en
el proceso
{
    digitalWrite(Bomba[1],LOW);
    digitalWrite(Solenoides[2],LOW);
    digitalWrite(SSR[1],LOW);
    N_Pag_Mensajes.show();
    nex.refresh();
    nex.sendCommand("t0.font=0");
    nex.sendCommand("vis t1,1");
    N_Str_Mensaje0.setText(mensaje_alerta_flujometro);
    nex.sendCommand("vis t2,1");
    nex.sendCommand("t2.font=0");
    N_Str_Mensaje2.setText(mensaje_alerta);
    Estado_Boton=0;
    while(Estado_Boton==0){
        presionar_boton();
    }
    HMI(Proceso);
    digitalWrite(Solenoides[2],HIGH);
    digitalWrite(Bomba[1],HIGH);
}
break;
case 4://Enfriado
//Revisar Flujometro
pulseCount=0;
attachInterrupt(digitalPinToInterrupt(flujometro[Que_Flujometro]),
pulseCounter, RISING);
delay(500);
detachInterrupt(flujometro[Que_Flujometro]);
if (pulseCount==0)
{
    Flujometro_Malo=TRUE;
    mensaje_alerta_flujometro="Error Flow/Bomba/Sol F.2";
    alert="F2-F2";
}else{Flujometro_Malo=FALSE;alert="OK";}
//Revisar Sensor Nivel
estado[0]=digitalRead(S_Nivel[2]);
estado[1]=digitalRead(S_Nivel[3]);
if(estado[0]==1 ||estado[1]==1)
{
    delay(300);
    estado[0]=digitalRead(S_Nivel[2]);
    estado[1]=digitalRead(S_Nivel[3]);
}
if (estado[0]==0 && estado[1]==0)//ALERTA
{
    mensaje_alerta="NO HAY AGUA EN EL FONDO 2";
}

```

```

        alert="F2-SN.Inf";
        Nivel_Malo=TRUE;
    }
    if (estado[0]==0 && estado[1]==1)//CASO IMPOSIBLE
    {
        mensaje_alerta="ARREGLAR SENSOR SUPERIOR FONDO 2";
        alert="F2-SN.Sup";
        Nivel_Malo=TRUE;
    }
    if (estado[0]==1 && estado[1]==0)//OK
    {
        mensaje_alerta="Niveles OK";
        if (alert!="F2-F2")
        {
            alert="OK";
        }
        Nivel_Malo=FALSE;
    }
    if (estado[0]==1 && estado[1]==1)//ALERTA
    {
        mensaje_alerta="FONDO 2 SE REBALSA";
        alert="F2-SN.Sup";
        Nivel_Malo=TRUE;
    }
    //Accion si ocurre error
    if (Nivel_Malo==TRUE || FlujoMetro_Malo==TRUE)//Hay algo malo en
    el proceso
    {
        digitalWrite(Bomba[1],LOW);
        digitalWrite(Solenoides[2],LOW);
        digitalWrite(Bomba[2],LOW);
        N_Pag_Mensajes.show();
        nex.refresh();
        nex.sendCommand("t0.font=0");
        nex.sendCommand("vis t1,1");
        N_Str_Mensaje0.setText(mensaje_alerta_flujometro);
        nex.sendCommand("vis t2,1");
        nex.sendCommand("t2.font=0");
        N_Str_Mensaje2.setText(mensaje_alerta);
        Estado_Boton=0;
        while(Estado_Boton==0){

            presionar_boton();

        }
        //reiniciar HMI
        HMI(Que_Proceso);
        digitalWrite(Solenoides[2],HIGH);
        digitalWrite(Bomba[1],HIGH); //Bomba recirculacion macerado
    }
    break;
    case 5://Traspaso
    //Revisar FlujoMetro
    if (pulseCount==0)
    {
        FlujoMetro_Malo=TRUE;
        mensaje_alerta_flujometro="Error Flow/Bomba/Sol F.2";
        alert="F2-F3";
    }

```

```

}else{Flujometro_Malo=FALSE;alert="OK";}
//Revisar Sensor Nivel
estado[0]=digitalRead(S_Nivel[2]);
estado[1]=digitalRead(S_Nivel[3]);
if(estado[0]==1 || estado[1]==1)
{
    delay(300);
    estado[0]=digitalRead(S_Nivel[2]);
    estado[1]=digitalRead(S_Nivel[3]);
}
if (estado[0]==0 && estado[1]==1)//CASO IMPOSIBLE
{
    mensaje_alerta="ARREGLAR SENSOR SUPERIOR FONDO 2";
    alert="F2-SN.Sup";
    Nivel_Malo=TRUE;
}
if (estado[0]==1 && estado[1]==0)//OK
{
    mensaje_alerta="Niveles OK";
    if (alert!="F2-F3")
    {
        alert="OK";
    }
    Nivel_Malo=FALSE;
}
if (estado[0]==1 && estado[1]==1)//ALERTA
{
    mensaje_alerta="FONDO 2 SE REBALSA";
    alert="F2-SN.Sup";
    Nivel_Malo=TRUE;
}
//Accion si ocurre error
if (Nivel_Malo==TRUE || Flujometro_Malo==TRUE)//Hay algo malo en
el proceso
{
    digitalWrite(Bomba[1],LOW);
    digitalWrite(Solenoides[3],LOW);
    digitalWrite(SSR[1],LOW);
    N_Pag_Mensajes.show();
    nex.refresh();
    nex.sendCommand("vis t2,1");
    nex.sendCommand("t2.font=0");
    N_Str_Mensaje2.setText(mensaje_alerta);
    nex.sendCommand("t0.font=0");
    nex.sendCommand("vis t1,1");
    N_Str_Mensaje0.setText(mensaje_alerta_flujometro);
    Estado_Boton=0;
    while(Estado_Boton==0){
        presionar_boton();
    }
    HMI(Proceso);
    digitalWrite(Solenoides[3],HIGH);
    digitalWrite(Bomba[1],HIGH);
    delay(2000);
}
break;

```

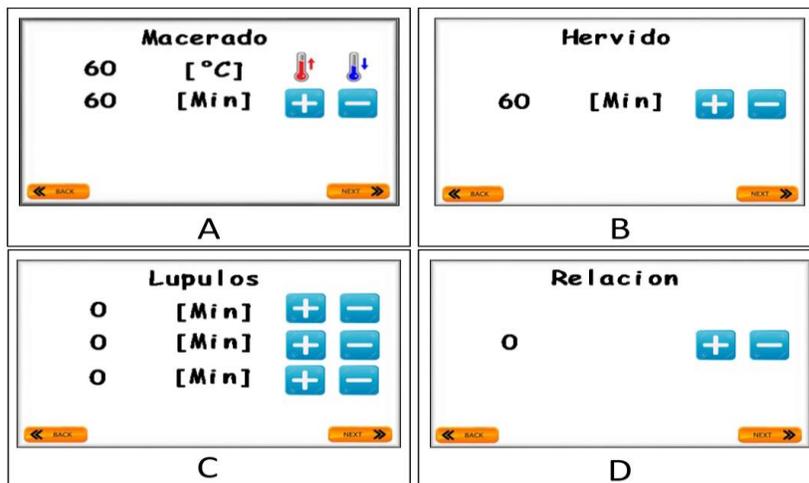
```
    }
    if (alert!="OK")
    {
        imprimir=set1 +";"+ temp1 +";"+out1+";"+set2 +";"+ temp2
+";"+out2+";"+Tiempo_Faltante+";"+Tiempo+";"+mensaje+";"+wort+";"+h2o+";"+Etapa+";"+ale
rt;//SD
        interval=10;
        sd(interval,imprimir);
    }
}
```

SOLO USO ACADÉMICO

## ANEXO N°2. Ilustraciones del HMI en diferentes etapas del proceso.

### 1. HMI parámetros

Ilustración 1 HMI Parámetros: Macerado (A), Hervido (B), Lúpulos (C), Relación Malta/Agua (D)



Fuente: Propia (2016)

### 2. HMI calentando

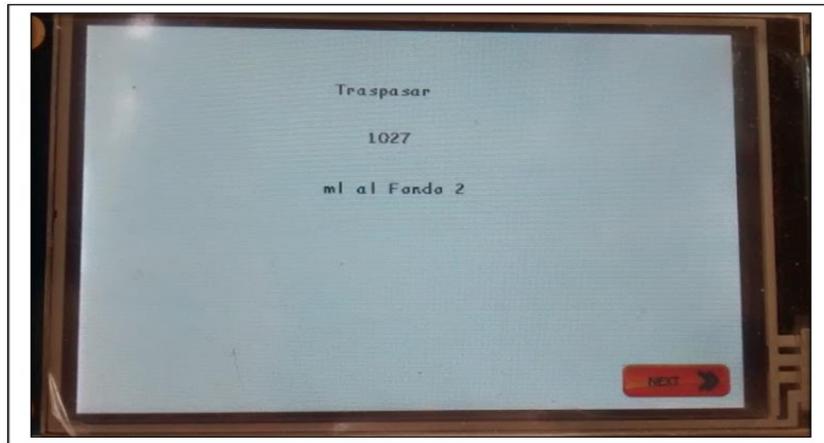
Ilustración 2 HMI Calentando



Fuente: Propia (2016)

### 3. HMI pre traspaso

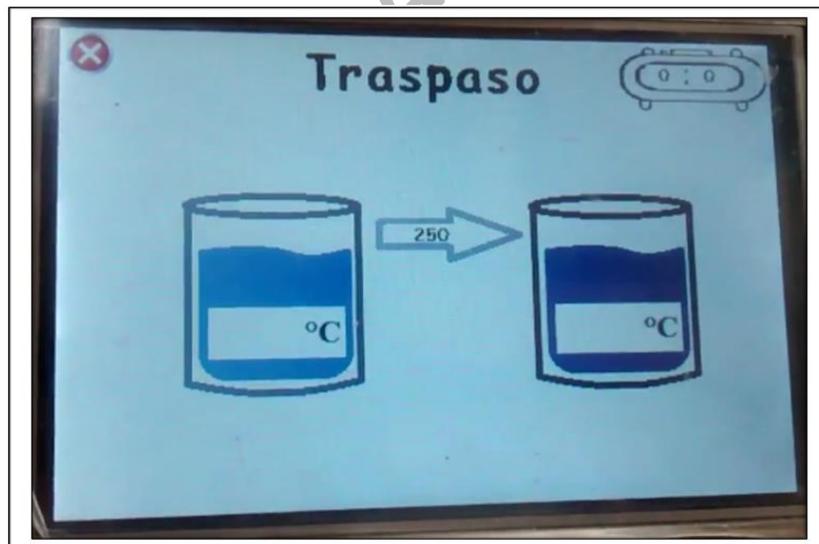
Ilustración 3 Pre Traspaso



Fuente: Propia (2016)

### 4. HMI traspaso

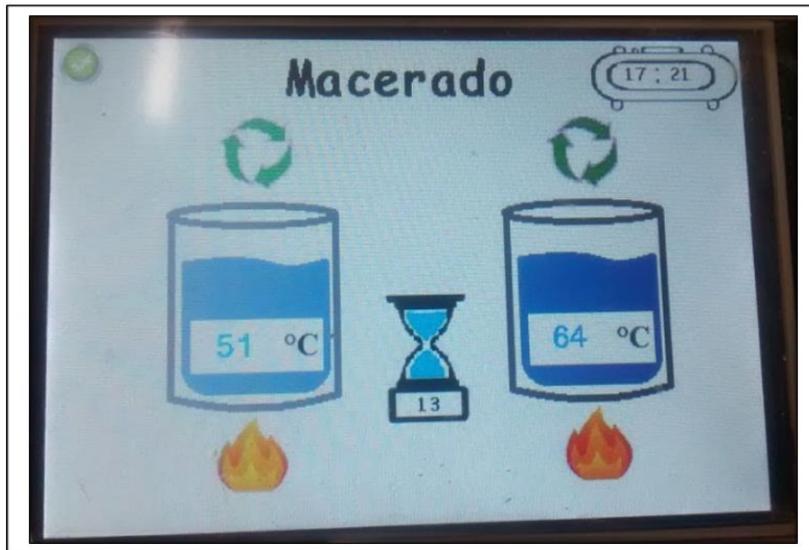
Ilustración 4 Traspaso



Fuente: Propia (2016)

## 5. HMI macerado

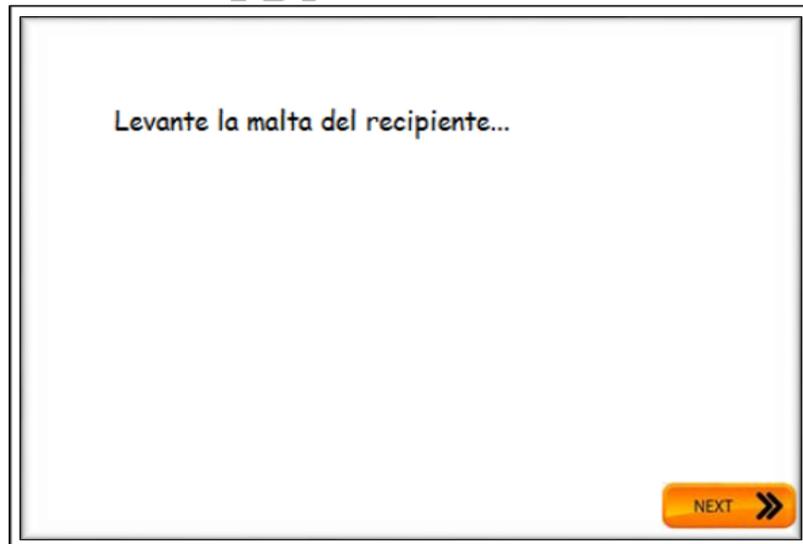
Ilustración 5 Macerado



Fuente: Propia (2016)

## 6. HMI levantar malta

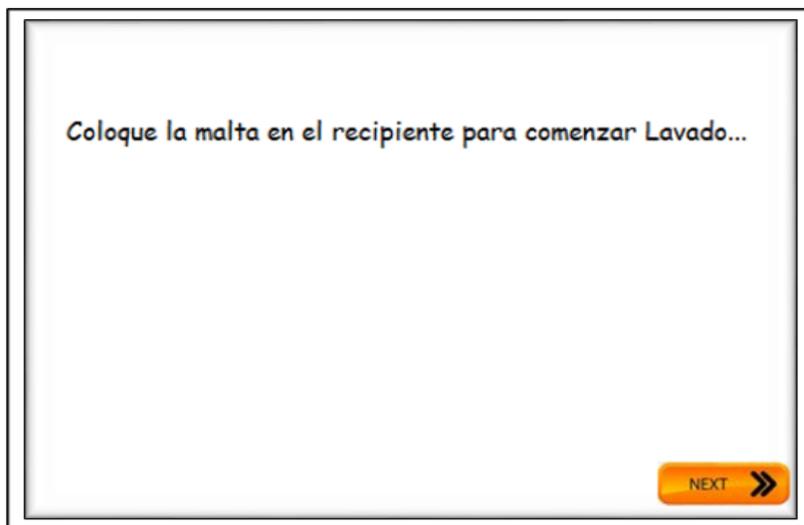
Ilustración 6 Levantar Malta



Fuente: Propia (2016)

## 7. HMI colocar malta para el lavado

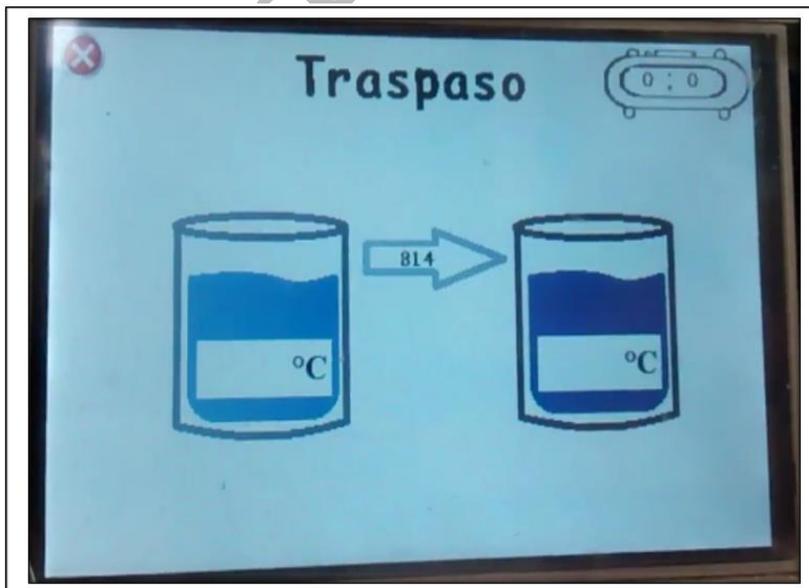
Ilustración 7 Colocar Malta



Fuente: Propia (2016)

## 8. HMI Traspaso para el lavado

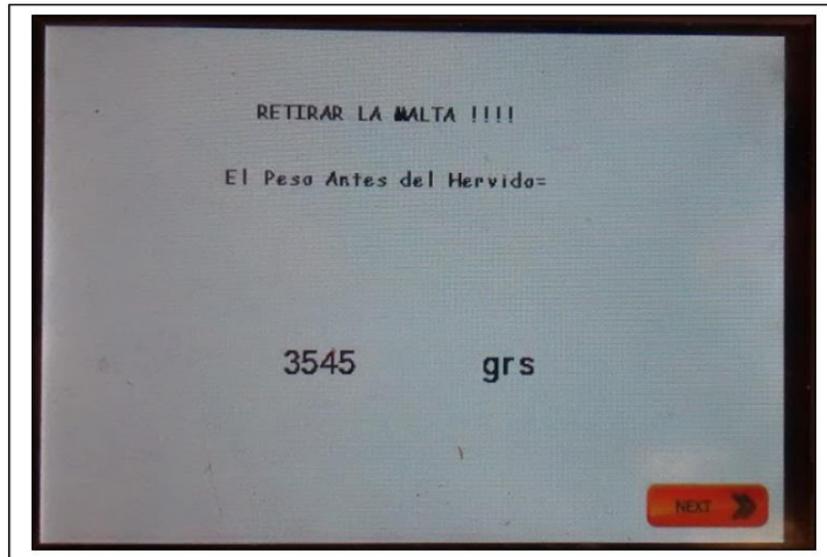
Ilustración 8 Traspaso Lavado



Fuente: Propia (2016)

## 9. HMI Retirar la malta

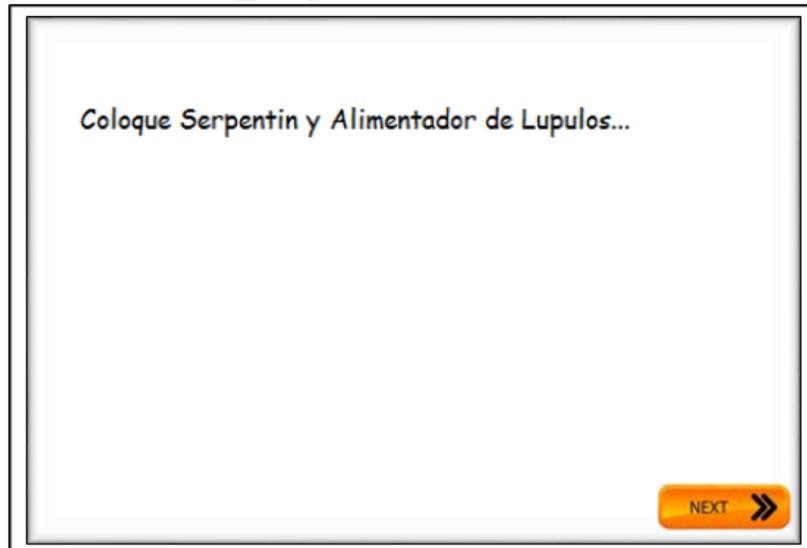
Ilustración 9 Retirar Malta



Fuente: Propia (2016)

## 10. HMI colocar serpentín y el alimentador de lúpulos

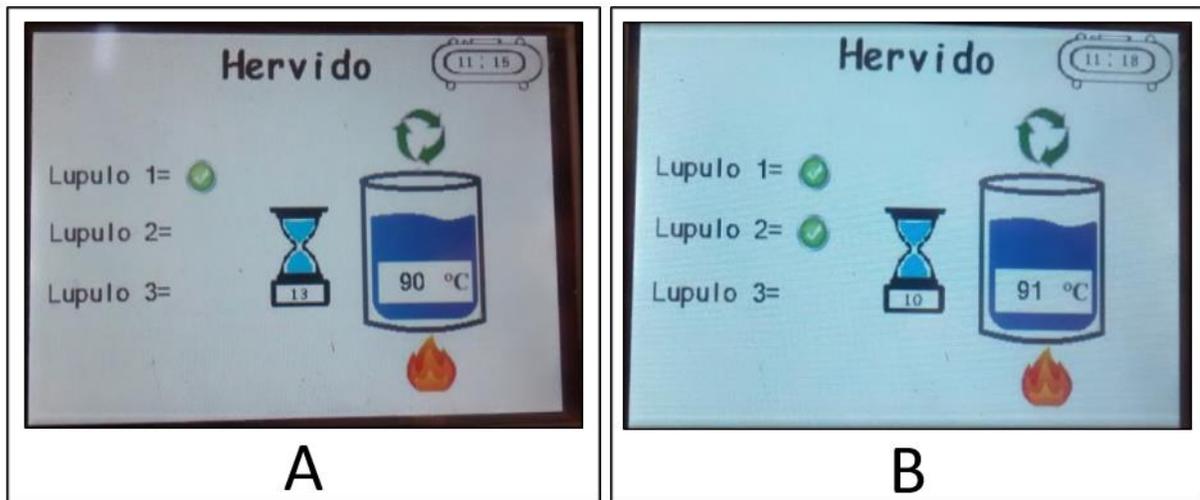
Ilustración 10 Serpentín y Alimentador lúpulos



Fuente: Propia (2016)

## 11.HMI hervido

Ilustración 11 Representación del primer lúpulo ingresado (A), Representación del segundo lúpulo ingresado (B)



Fuente: Propia (2016)

## 12.HMI enfriado

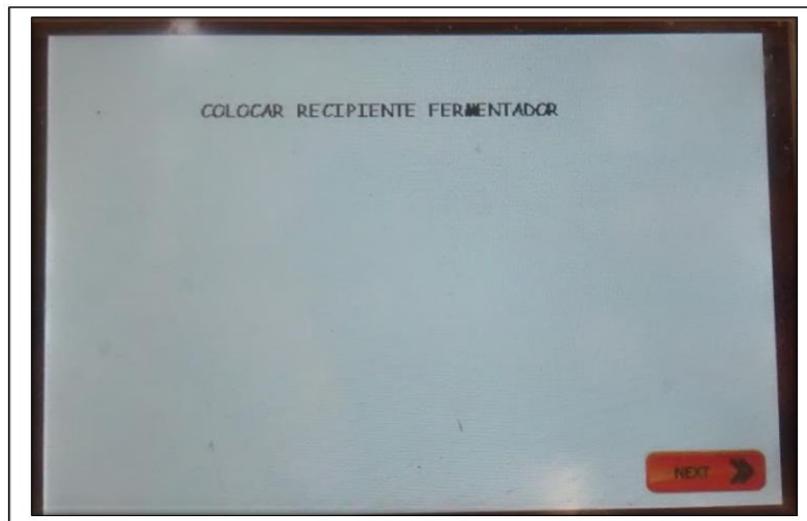
Ilustración 12 Enfriado



Fuente: Propia (2016)

### 13.HMI colocar recipiente fermentador

Ilustración 13 Colocar recipiente fermentador



Fuente: Propia (2016)

### 14.HMI traspaso al recipiente fermentador

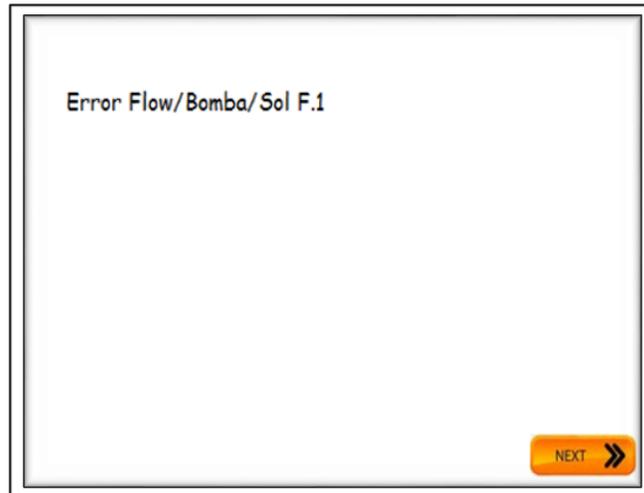
Ilustración 14 Traspaso recipiente fermentador



Fuente: Propia (2016)

## 15.HMI Alerta

*Ilustración 15 Alerta de error del solenoide, bomba o fluxómetro 1. Se encuentra en el primer recipiente*



*Fuente: Propia (2016)*

SOLO USO ACADEMICO

### ANEXO N°3. Montaje de los diferentes actuadores y sensores.

#### 1. Montaje sensor de nivel

Ilustración 16 Montaje Sensor de Nivel



Fuente: Propia (2016)

#### 2. Montaje válvulas de solenoides

Ilustración 17 Montaje sensor de temperatura, válvulas solenoides con sus respectivos fluxómetros



Fuente: Propia (2016)

### 3. Montaje alimentador de lúpulos

*Ilustración 18 Montaje Alimentador de Lúpulos en recipiente 2*



*Fuente: Propia (2016)*

### 4. Montaje serpentín de cobre

*Ilustración 19 Montaje serpentín de cobre en el recipiente 2*



*Fuente: Propia (2016)*

## 5. Montaje Sensor de Peso

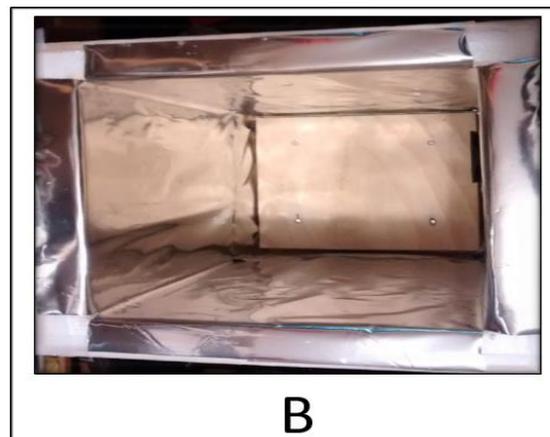
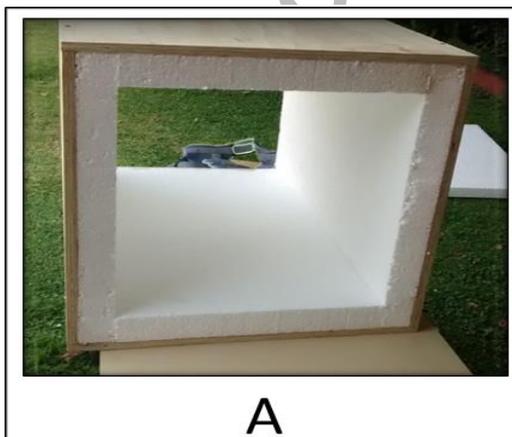
Ilustración 20 Montaje Sensor de Peso en el carrito



Fuente: Propia (2016)

## 6. Cajón fermentador

Ilustración 21 Cajón fermentador: Aislación Plumavit (A), Aislación Aluminizado (B)



Fuente: Propia (2016)

## 7. Montaje Sensor Alcohol

*Ilustración 22 Pieza impresa de montaje sensor alcohol*



*Fuente: Propia (2016)*

## 8. Montaje sistema completo

*Ilustración 23 Carrito con todos los componentes montados*



*Fuente: Propia (2016)*

## ANEXO N°4. Hoja de datos de los sensores y actuadores del sistema.

### 1. Hoja de datos sensor temperatura

#### DS18B20

#### Programmable Resolution 1-Wire Digital Thermometer

##### General Description

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

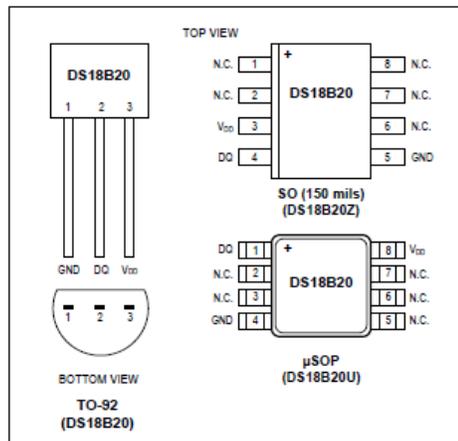
##### Applications

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

##### Benefits and Features

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Reduce Component Count with Integrated Temperature Sensor and EEPROM
  - Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
  - ±0.5°C Accuracy from -10°C to +85°C
  - Programmable Resolution from 9 Bits to 12 Bits
  - No External Components Required
- Parasitic Power Mode Requires Only 2 Pins for Operation (DQ and GND)
- Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability
  - Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits
- Available in 8-Pin SO (150 mils), 8-Pin  $\mu$ SOP, and 3-Pin TO-92 Packages

##### Pin Configurations



[Ordering Information](#) appears at end of data sheet.

1-Wire is a registered trademark of Maxim Integrated Products, Inc.



## 2. Hoja de datos sensor de peso

### INFO

La medición del peso es muy importante en diversos procesos industriales, médicos, de laboratorio. Es por eso que es necesario contar con celdas de carga capaces de convertir la medida en peso en una señal electrónica.

Esta celda de 20Kg posee una excelente precisión y buen desempeño, fabricado en aluminio y con agujeros para montaje.

Tiene como salida 4 cables, que funcionan como un puente wheatstone, por lo que para poder realizar las lecturas mediante un microcontrolador es necesario acondicionar la señal. El transmisor Hx711 está especialmente diseñado para trabajar con celdas de carga de este tipo y ofrecer una salida en voltaje, la cual puede ser conectada a las entradas analógicas del microcontrolador.

### ESPECIFICACIONES TÉCNICAS

- Modelo: FZ1439
- Capacidad de carga nominal: 20 Kg
- Salida de Cero: 0.05% FS
- Precisión: 0.02% FS
- Precisión-Temperatura: 0.02% FS
- Temperatura de trabajo: -10°C a 50°C
- Sobre carga de seguridad: 150%
- Salida: 4 cables (puente wheatstone)
- Dimensiones: 80x12.7x12.7mm
- Agujeros (4): D3.4mm
- Longitud cables: 16 cm
- Material: Aluminio
- Peso: 31g

SOLO USO ACADÉMICO

### 3. Hoja de datos conversor análogo digital

## 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales

### DESCRIPTION

Based on Avia Semiconductor's patented technology, HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of  $\pm 20\text{mV}$  or  $\pm 40\text{mV}$  respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization.

There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

### FEATURES

- Two selectable differential input channels
- On-chip active low noise PGA with selectable gain of 32, 64 and 128
- On-chip power supply regulator for load-cell and ADC analog power supply
- On-chip oscillator requiring no external component with optional external crystal
- On-chip power-on-reset
- Simple digital control and serial interface: pin-driven controls, no programming needed
- Selectable 10SPS or 80SPS output data rate
- Simultaneous 50 and 60Hz supply rejection
- Current consumption including on-chip analog power supply regulator:
  - normal operation  $< 1.5\text{mA}$ , power down  $< 1\mu\text{A}$
- Operation supply voltage range: 2.6 ~ 5.5V
- Operation temperature range:  $-40 \sim +85^\circ\text{C}$
- 16 pin SOP-16 package

### APPLICATIONS

- Weigh Scales
- Industrial Process Control

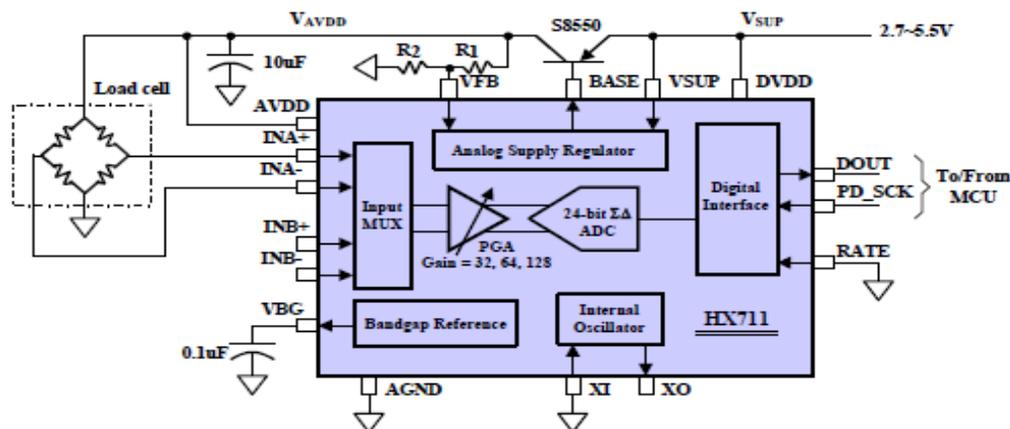


Fig. 1 Typical weigh scale application block diagram

#### 4. Hoja de datos sensor de etanol

## TECHNICAL DATA MQ-3 GAS SENSOR

### FEATURES

- \* High sensitivity to alcohol and small sensitivity to Benzine .
- \* Fast response and High sensitivity
- \* Stable and long life
- \* Simple drive circuit

### APPLICATION

They are suitable for alcohol checker, Breathalyser.

### SPECIFICATIONS

#### A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V <sub>c</sub>	Circuit voltage	5V±0.1	AC OR DC
V <sub>H</sub>	Heating voltage	5V±0.1	AC OR DC
R <sub>L</sub>	Load resistance	200KΩ	
R <sub>H</sub>	Heater resistance	33Ω ± 5%	Room Tem
P <sub>H</sub>	Heating consumption	less than 750mW	

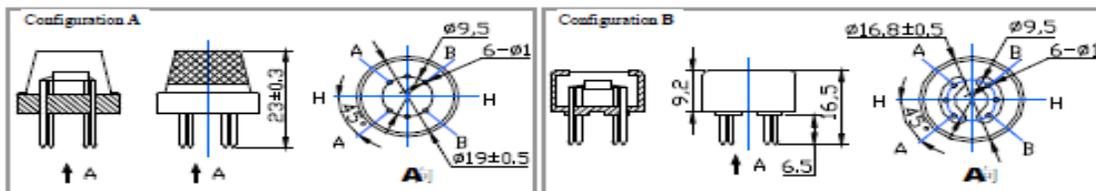
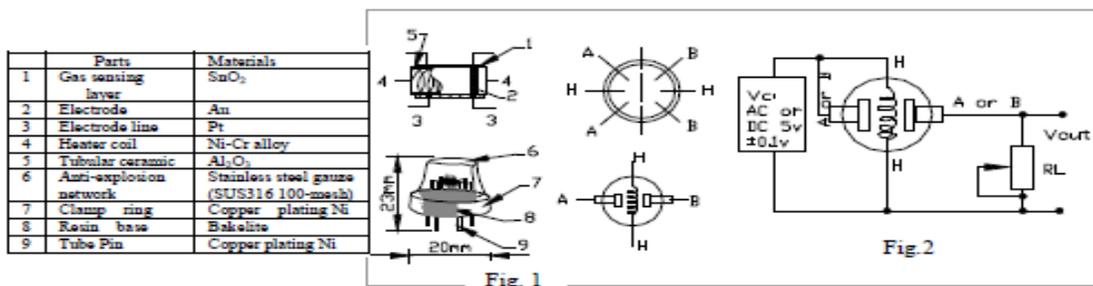
#### B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
T <sub>ao</sub>	Using Tem	-10℃-50℃	
T <sub>as</sub>	Storage Tem	-20℃-70℃	
R <sub>H</sub>	Related humidity	less than 95%Rh	
O <sub>2</sub>	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

#### C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
R <sub>s</sub>	Sensing Resistance	1MΩ - 8MΩ (0.4mg/L alcohol )	Detecting concentration scope: 0.05mg/L—10mg/L Alcohol
α (0.4/1 mg/L)	Concentration slope rate	≤0.6	
Standard detecting condition	Temp: 20℃ ± 2℃ Humidity: 65%±5%	V <sub>c</sub> :5V±0.1 V <sub>H</sub> : 5V±0.1	
Preheat time	Over 24 hour		

#### D. Structure and configuration, basic measuring circuit



Structure and configuration of MQ-3 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro  $Al_2O_3$  ceramic tube, Tin Dioxide ( $SnO_2$ ) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-3 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

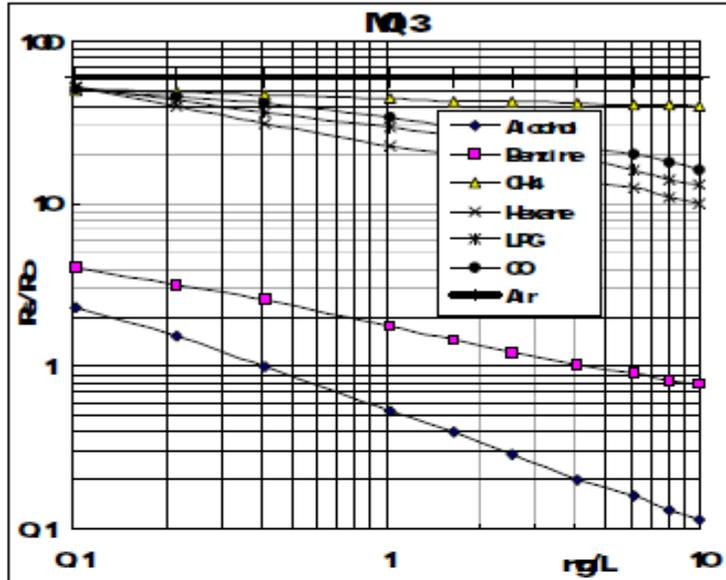


Fig 2 sensitivity characteristics of the MQ-3

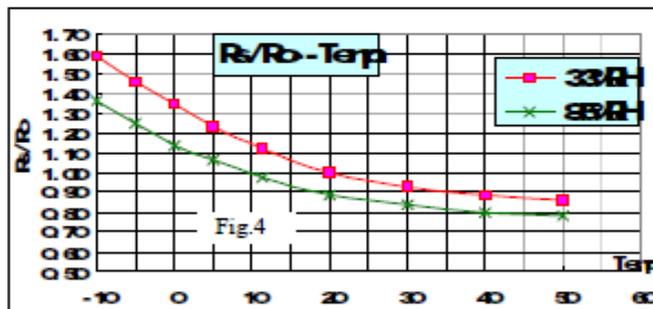


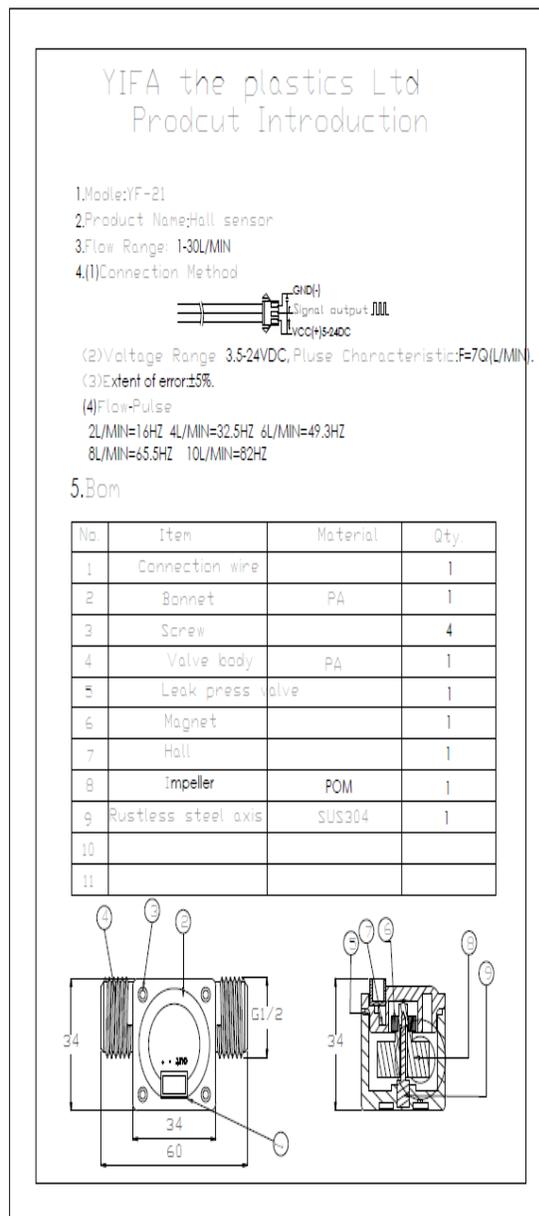
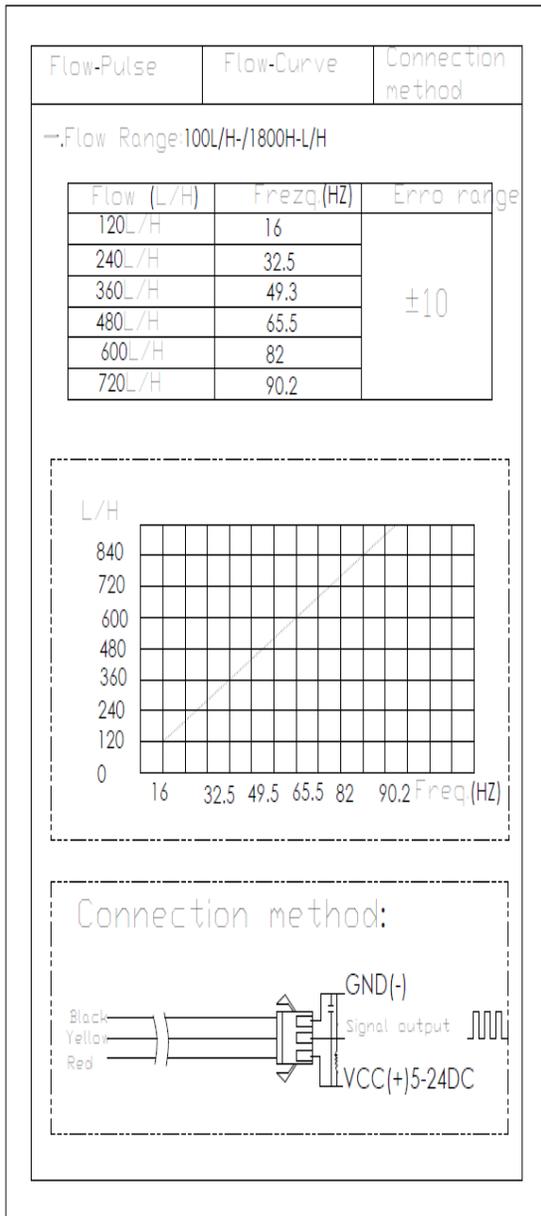
Fig.4 is shows the typical dependence of the MQ-3 on temperature and humidity.  
 $R_0$ : sensor resistance at 0.4mg/L of Alcohol in air at 33%RH and 20 °C  
 $R_s$ : sensor resistance at 0.4mg/L of Alcohol at different temperatures and humidities.

### SENSIVITY ADJUSTMENT

Resistance value of MQ-3 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 0.4mg/L (approximately 200ppm) of Alcohol concentration in air and use value of Load resistance that ( $R_L$ ) about 200 K $\Omega$  (100K $\Omega$  to 470 K $\Omega$ ).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

## 5. Hoja de datos sensor de flujo



## 6. Hoja de datos sensor de nivel

Color: plata

Material: acero inoxidable SUS

Contactos Max: 10 W

Max conmutación Voltaje: 100 V DC

Max Corriente de conmutación: 0.5A

Max tensión: 220 V DC

Max Corriente de transporte: 1A

Máxima resistencia del contacto: 100mw

Temperatura máxima: -30 y #8451; -+ 125 y #8451;

Tamaño de rosca (aprox): 10mm

Cambio Longitud total: 61mm/2.4"

Longitud del cable: aprox 350mm/13.78"

### **Características:**

Este es un mini interruptor de flotador de acero inoxidable, no contiene mercurio.

Peso ligero y principio de funcionamiento simple, alta confiabilidad.

Este interruptor de flotador es un dispositivo utilizado para detectar el nivel de líquido dentro de un tanque,

El interruptor puede activar una bomba, un indicador, una alarma u otro dispositivo.

Este interruptor de flotador es una estructura simple, fácil Instalación controlador de nivel de líquido.

No tiene un circuito complejo, no será perturbado,

Siempre y cuando la selección de materiales, líquido, presión o temperatura se pueden utilizar en cualquier entorno.

La versatilidad de este sensor verticalmente monted significa que

Puede ser instalado en la parte superior o inferior de su tanque,

Dependiendo de si usted para subir o bajar de deseo para detectar el agua.

El interruptor de nivel de flotador es ampliamente utilizado en electrónica, eléctrica, química,

Tratamiento de agua, drenaje y otros ámbitos de la Vida Control de nivel y alarma.



## 7. Hoja de datos válvula de solenoide

### Description

DESCRIPTION

#### Descripción:

Reparación de una válvula defectuosa con esta alta calidad, de acción rápida 12 V DC reemplazo de estilo de diafragma que es adecuado para su uso en aplicaciones de agua y fluidos de baja viscosidad.

#### Características:

A estrenar y alta calidad.

Material: metal + plástico.

Voltaje: 12 V DC.

Tipo de válvula: diafragma (Operado por servo).

Modo de funcionamiento: normalmente cerrado.

Temperatura del fluido: 1-100 grados.

Tamaño: 8.5 cm x 6 cm x 4.3 cm.

SOLO USO ACK

## 8. Hoja de datos bomba alimentaria

### Item specifics

Power: Electric

Pressure: High Pressure

Theory: Electromagnetic Pump

Application: Pour liquid

Usage: beverage

Power: 15W

Maximum Pump lift: 5m

Material: Engineering Plastics

Self-priming: No

Power Supply: Batteries, power modules, solar

Structure: Diaphragm Pump

Standard or Nonstandard: Standard

Model Number: Pump01

Fuel: Electromagnetic drive

Voltage: 12V

Maximum flow: 10L/min

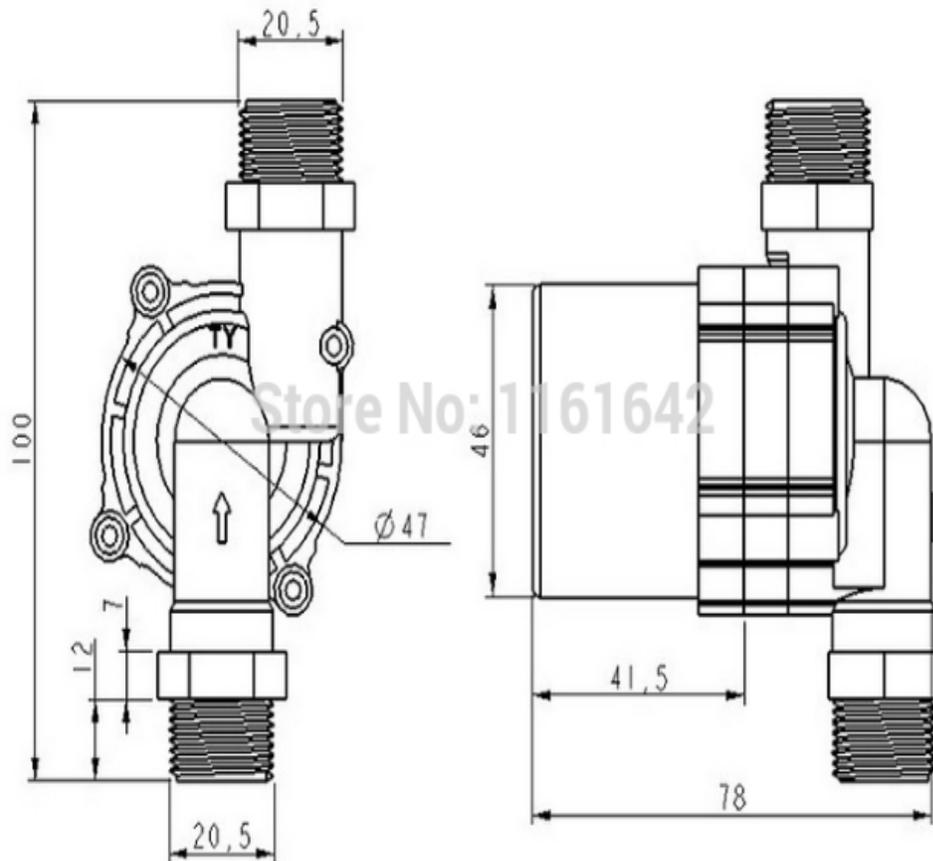
Rated temperature: 110°C (230°F)

Electric current: 1.5A

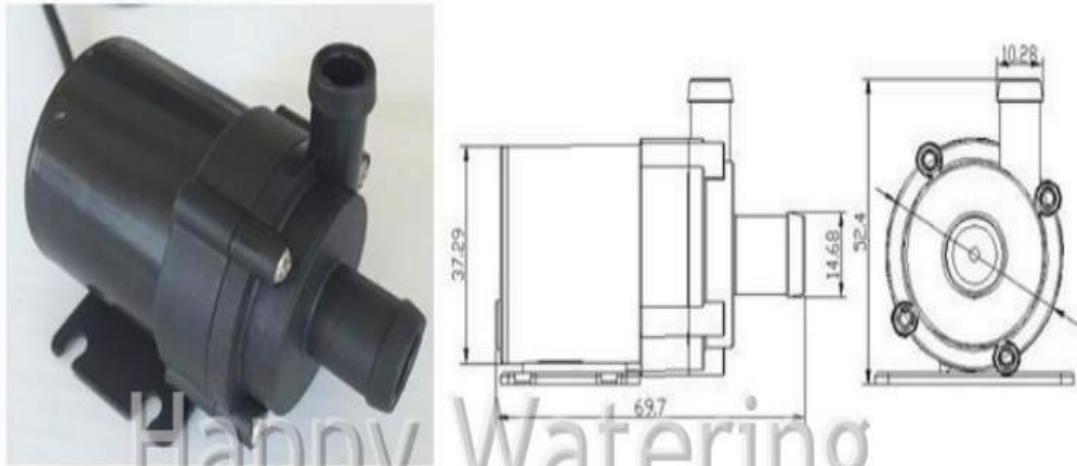
Pump idling: Not allowed

### Product Description

#### Food grade hot water brew pump and circulation brewing pump DC12V&20W



## 9. Hoja de datos bomba de agua serpentín de cobre



Model 型号	Voltage Scope (DC) 电压	Current (A) 电流	Power (W) 功率	Max Water Head (M) 扬程	Max Flow Rate (L/H) 流量	Starting Voltage 启动电压	waterproo fing grade 防水等级
JT-600A-12-3	12V	0.6	7.2	2.5	400	5	IP68
JT-600A-12	12V	0.85	10	3.8	600	5	IP68
JT-600A-24	24V	1	24	5	700	6	IP68

What we sell here is 600A-12 type. If need 24v, just giving us an message.

### Packages:

- 1 X water pump

### Specifications:

- New Mini Water Pump 12V DC Solar Fountain Pump
- Housing material: Tough ABS
- Voltage: 12V
- Max Output: 600L/h
- Max Lift: 3.8m
- Stainless steel shaft
- Pressure: 0.5bar
- Life Span: about 30000 hrs
- Working: Continuous
- Cable length: 0.5m
- Heat Resistance: -40°C-100°
- Self-priming: No
- Noise: 40db MAX
- The pump can be used underwater diving and land type.
- The brown line connect to "+", The blue line connect to "-".

10. Hoja de datos elemento calentador

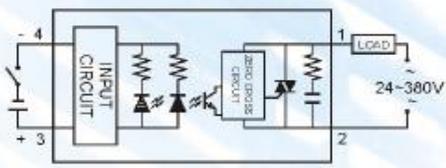
**Stainless Steel Electric Kettle Element Heating Tube Heater AC 220V 3KW**  
**Specification:**

Product Name	Electric Kettle Element
Material	Stainless Steel
Rated Voltage	AC 220V
Power	3KW
Overall Size (Approx.)	12 x 6cm/ 4.72" x 2.36" (D*H)
Main Color	Silver Tone
Weight	200g
Package Content	1 x Electric Kettle Element

SOLO USO ACAL

## 11. Hoja de datos SSR

### ■ Specification

Type	Terminal Type					PCB Type
Model	<b>SSR-10DA</b>	<b>SSR-25DA</b>	<b>SSR-40DA</b>	<b>SSR-25DA-H</b>	<b>SSR-40DA-H</b>	<b>SSR-P03DA</b>
Rated Load Current	<b>10A</b>	<b>25A</b>	<b>40A</b>	<b>25A</b>	<b>40A</b>	<b>3A</b>
<b>Input Data</b>						
Operating Voltage	3~32VDC					
Min. ON / OFF Voltage	ON > 2.4V , OFF < 1.0V					
Trigger Current	7.5mA / 12V					
Control Method	Zero Cross Trigger					
<b>Output Data</b>						
Operating Voltage	24~380VAC		90~480VAC		24~380VAC	
Min. Block Voltage	600 VAC < Repetive >					
Voltage Drop	1.6 V / 25 C					
Max. Durated Current	135A	275A	410A	275A	410A	135A
Leakage Current	3.0mA	3.0mA	3.0mA	5.0mA	5.0mA	3.0mA
Response Time	ON < 10ms , OFF < 10ms					
<b>General Data</b>						
Dielectric Strength	Over 2.5KVAC / 1min.					
Isolation Strength	Over 50MΩ / 500VDC					
Operating Temperature	-20 C ~+80 C					
Housing Material	Intensive ABS					
Weight	Appr. 105g					Appr. 15g
<b>Connection Diagram</b>						
						

## 12. Hoja de datos pantalla táctil

### Caution:

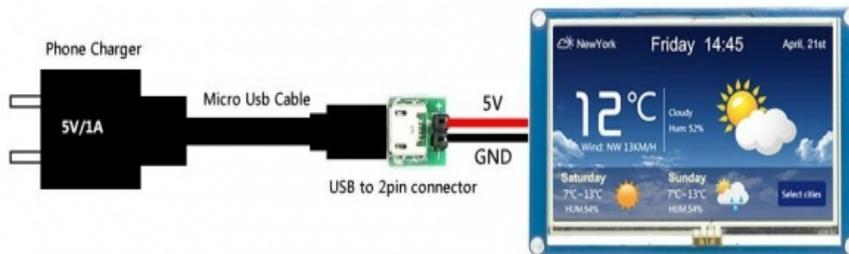
Working under insufficient power supply condition will damage the Nextion model easily.



Blurred screen? Flashing? You may suffer from power shortages. Power off at the first possible moment, NO MORE repeated trying to damage your Nextion model.

A small connector is included in the package, please try to power Nextion with your phone charger through the connector to test if Nextion works well.

A high quality usb cable is required.



Nextion User Manual: <http://goo.gl/LbvAJ5>

## Specification

	Data	Description
Color	65K (65536) colors	16 bit, 5R6G5B
Layout size	100.5 (L)×54.94 (W)×4.25 (H)	NX4832T035_011N
	100.5 (L)×54.94 (W)×5.45 (H)	NX4832T035_011R
Active Area (A.A.)	85.50mm(L)×54.94mm(W)	-
Visual Area (V.A.)	73.44mm(L)×48.96mm(W)	-
Resolution	480×320 pixel	Also can be set as 320×240
Touch type	Resistive	-
Touches	> 1 million	-
Backlight	LED	-
Backlight lifetime (Average)	>30,000 Hours	-
Brightness	200nit (NX4832T035_011N)	0% to 100%, the interval of adjustment is 1%
	180 nit (NX4832T035_011R)	0% to 100%, the interval of adjustment is 1%
Weight	38.2g (NX4832T035_011N)	-
	48.2g (NX4832T035_011R)	-

## ANEXO N°5. Datos diversos que respaldan algunos cálculos realizados.

### 1. Datos del sensor de alcohol en prueba del fermentador

Ilustración 24 Datos sensor de alcohol en prueba del fermentador

Día	Dato N°	Promedio Diario	Variación con el día anterior
1	1440	264	-
2	2880	393	49%
3	4320	541	38%
4	5760	607	12%
5	7200	634	5%
6	8640	625	2%
7	10080	602	4%
8	11520	591	2%

Datos Sensor Alcohol

Fuente: Propia (2016)

## INDICE DE TABLAS

Tabla 1 Análisis de Inversiones y Costos ----- 72

SOLO USO ACADÉMICO

## INDICE DE FIGURAS

Figura 1 Brewie-----	13
Figura 2 Brew-Boss-----	14
Figura 3 Grainfather-----	15
Figura 4 Diagrama en bloques simplificado del sistema -----	18
Figura 5 Diagrama de flujo simplificado del controlador -----	20
Figura 6 Diagrama de Bloques-----	21
Figura 7 Función Parámetros -----	24
Figura 8 Función Calentando-----	26
Figura 9 Función Macerado -----	28
Figura 10 Función Lavado-----	30
Figura 11 Función Hervido-----	32
Figura 12 Función Enfriado -----	34
Figura 13 Función Traspaso Fermentador -----	36
Figura 14 Función Fermentador-----	38
Figura 15 Función Proceso-----	40
Figura 16 Función Temp Estable -----	42
Figura 17 Función PID-----	43
Figura 18 Función Datos -----	45
Figura 19 Función Alerta -----	47
Figura 20 Estructura del Sistema-----	50
Figura 21 Controlador Arduino Mega 2560 -----	51
Figura 22 Sensor DS18B20-----	52
Figura 23 Celda de Peso FZ1439 -----	53
Figura 24 MQ3 sensor -----	54
Figura 25 Fluxómetro YF-S201-----	55
Figura 26 Sensor Nivel Exterior (A), Sensor Nivel Interior (B)-----	56
Figura 27 Válvula Solenoide-----	57
Figura 28 Bomba Alimentaria (A), Bomba Alimentaria Interior (B), Bomba Enfriamiento (C) -----	58
Figura 29 Elemento Calentador (A), SSR (B)-----	59
Figura 30 Disco Superior (A), Disco Inferior (B), Soporte Motor (C), Polea Motor (D) -----	61
Figura 31 HMI-----	62
Figura 32 Comportamiento de la Planta (A), Función de Transferencia de la Planta (B), Parámetros del Controlador de los Elementos Calentadores (C), Prueba realizada al Calentador (D) -----	65
Figura 33 Parámetros del Controlador del Fermentador (A), Prueba Realizada al Fermentador (B) --	66
Figura 34 Prueba Sensor de Alcohol -----	67
Figura 35 Prueba Perdida de Agua mediante el monitoreo de su peso -----	68
Figura 36 Protocolo establecido al momento de guardar los datos-----	69
Figura 37 Prueba del Sistema Completo -----	70