



**UNIVERSIDAD  
MICHUACANA DE SAN  
NICOLÁS DE HIDALGO**



**FACULTAD DE INGENIERÍA ELÉCTRICA**

**“AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN”**

**TESIS**

**QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN ELECTRÓNICA**

**PRESENTA:  
EFRÉN ZÚÑIGA CHÁVEZ**

**ASESOR:  
INGENIERO ELECTRICISTA  
IGNACIO FRANCO TORRES**

Morelia, Mich.

Mayo 2020

# Agradecimientos

Gracias principalmente a Dios por permitirme cumplir este gran sueño y seguir creciendo como persona y profesional ante los retos que significa alcanzar esta meta.

La realización de este trabajo no hubiese sido posible sin la ayuda incondicional de personas que durante el proceso aportaron datos con su ayuda.

Agradezco a mi tío Salvador Zuñiga Mandujano, el cual fue el inspirador para que esta tesis fuera llevada a cabo y por su apoyo, emocional y económicamente para que este proyecto se implementara en la realidad.

Agradezco también al señor Rafael por su colaboración.

Agradezco a mi asesor Ignacio Franco Torres por su dedicación, constancia y creer que esto podía ser posible, su tiempo impartido en esta tesis y su conocimiento transmitido.

# Dedicatoria

Esta tesis se la dedico en especial a mi madre Martha Chávez Martínez, que siempre me apoyo y siempre dio lo mejor para mí, por haberme formado sin importar los obstáculos por los que tuvo que pasar, por ser la mujer que me dio la vida y me enseñó a vivirla, como también es mi gran orgullo y admiración de que las cosas son posibles siempre y cuando uno se lo propone y le dedica el tiempo necesario, y que; siempre uno puede terminar sus sueños cuando se lo propone.... Gracias mamá.

Esta tesis se la dedico a mi padre Efrén Zúñiga Mandujano, el cual siempre me ha enseñado lo mejor de la vida, aconsejándome sobre lo mejor para mi desarrollo, agradezco mucho sus consejos, su apoyo que a pesar de las dificultades siempre me ha escuchado, entendido y orientado.

Esta tesis se la dedico a mis hermanos Myrna Zúñiga Chávez y Abraham Zúñiga Chávez, que han estado en toda esta etapa de mi vida, que luchen por sus sueños y consigan lo que quieran que no les digan que no se puede, porque si se puede.

Esta tesis se la dedico con mucho cariño a mi novia Rosa Lidia Cisneros Aguilar la cual ha estado en toda esta etapa de mi vida apoyándome, creyendo en mí y animándome que esta tesis podría ser posible y que todo lo que me proponga lo puedo lograr a hacer. Muy agradecido porque fue parte de esta tesis, por su aportación y por su gran apoyo incondicional, con quien he compartido una gran parte de mi vida ya que ha sido compañera y amiga indispensable, fuente de calma y consejo en todo momento quien ha sido el impulso durante el desarrollo de mi carrera; muchas gracias te amo chaparra .

# Índice

Agradecimientos .....	ii
Dedicatoria.....	iii
Índice.....	iv
Resumen .....	vi
Palabras clave .....	vi
Abstract.....	vii
Keywords.....	vii
Lista de figuras .....	viii
Lista de tablas .....	ix
Glosario de términos .....	x
Capítulo 1 Introducción .....	1
1.1. Antecedentes .....	1
1.2. Proceso de nixtamalización.....	2
1.3. Objetivo general.....	3
1.4. Justificación .....	3
1.5. Metodología .....	3
1.6. Contenido de la Tesis .....	4
Capítulo 2 Estudio del proceso de la Nixtamalización.....	5
2.1. Estudio general de nixtamalización y planteamiento del algoritmo. ....	5
2.2. Estudio de tiempo del proceso de nixtamalización .....	8
2.3. Estudio de temperatura del proceso de nixtamalización .....	12
2.4. Estudio de pH del agua nixtamalizada .....	13
2.5. Maíz en reposo y nixtamalizado .....	14
Capítulo 3 Automatización del proceso de nixtamalización.....	16
3.1. Introducción .....	16
3.2. Sensores y Actuadores del sistema.....	16
3.3. El Arduino Uno como elemento de control .....	18
3.3.1. Hardware .....	19

3.3.2. Software.....	19
3.4. Aplicación en el celular para control del sistema .....	21
3.5. Algoritmo de control .....	21
Capítulo 4 Sensores y actuadores del sistema a controlar .....	22
4.1. Introducción .....	22
4.2. Arduino Uno .....	22
4.2.1. Componentes.....	25
4.3. Módulo de comunicación del bluetooth.....	34
4.4. Aplicación en App inventor .....	35
4.4.1. App Inventor .....	35
4.4.2. Desarrollo del Código en la plataforma Arduino .....	41
Capítulo 5 Prototipo del sistema a desarrollar .....	54
5.1. El Arduino como elemento de control.....	54
5.1.1. Descripción del control .....	54
5.1.2. Diagrama de flujo del software del microcontrolador. ....	55
5.2. Aplicación para control de Arduino .....	55
5.2.1. Descripción de la App .....	57
5.2.2. Diagrama de flujo del Software de la App.....	58
Capítulo 6 Prototipo desarrollado, pruebas y resultados.....	59
6.1. Esquemático.....	59
6.2. Pruebas y resultados.....	62
Capítulo 7 Conclusiones, recomendaciones y trabajos futuros. ....	63
7.1. Conclusiones .....	63
7.2. Recomendaciones .....	63
7.3. Trabajos futuros .....	64
Bibliografía .....	65

## Resumen

El prototipo automatiza el proceso de la nixtamalización, controlando los tiempos de cada una de las etapas del proceso de cocción; con la finalidad de obtener siempre el mismo resultado con mayor eficiencia de la caldera. El sistema de control se llevó a cabo mediante un lenguaje de programación, el cual es implementado en una placa de **Arduino Uno**, este se programó para realizar el control del proceso de nixtamalización al elegir la operación deseada mediante el desarrollo de una aplicación en Android. Para elegir el diseño de control más adecuado, se observaron las distintas formas de cocción, tanto su nivel de agua, cantidad de cal y el pH del agua ya nixtamalizada. Con esto se realiza el proceso de nixtamalización, sin la continua verificación del personal durante el proceso.

## Palabras clave

- Control, Facilidad de uso, Tiempo y Calidad.

# Abstract

The prototype automates the nixtamalización process, controlling the times of each of the stages on the cooking process; this in order to always obtain the same result and greater efficiency of the boiler. It was carried out through a programming language, which will be implemented in an Arduino Uno board, this was programmed to control the nixtamalización process by choosing the desired operation through the Android application. To choose the most appropriate design, different cooking methods were observed, both their water level, amount of lime and the pH of the water already nixtamalizado. With this, the procedure is carried out by means of the prototype, without the continuous verification of the personnel during the nixtamalización process.

## Keywords

- Check, easy to use, time and quality.

# Lista de figuras

Figura 1. Maíz puro y maíz nixtamalizado [1].....	1
Figura 2. Diagrama de flujo del proceso de nixtamalización.....	6
Figura 3. Partes que conforman el maíz [6]. .....	7
Figura 4. Molino industrial [4] .....	8
Figura 5. Molino de metate [1] .....	8
Figura 6. Caldera para la nixtamalización [7]. .....	10
Figura 7. Paso número 1 del proceso para la nixtamalización [7]. .....	11
Figura 8. Paso número 2 del proceso para la nixtamalización [7]. .....	12
Figura 9. Paso número 3 del proceso para la nixtamalización [7]. .....	13
Figura 10. Paso número 4 del proceso para la nixtamalización [7]. .....	13
Figura 11. Equipo medidor de pH [8] .....	14
Figura 12. Maíz recién Nixtamalizado [7]. .....	15
Figura 13. Maíz Nixtamalizado en proceso de reposo [7]. .....	15
Figura 14. Arduino Uno descripción de componentes [21] .....	23
Figura 15. Diagrama de bloques del Termómetro digital DS18B20.....	25
Figura 16. Gráfica de errores con relación a la temperatura [23]. .....	26
Figura 17. Termómetro DS18B20 [22]. .....	26
Figura 18. Sensor Ultrasónico [7]. .....	27
Figura 19. Rango de medición sensor HC-SR04 [25]. .....	28
Figura 20. Electroválvula solenoide para agua [21]. .....	28
Figura 21. Válvula solenoide para Gas LP [26].....	30
Figura 22. Diagrama de conexión del módulo generador de chispa.....	30
Figura 23. Modulo generador de chispa 15kv [7]. .....	31
Figura 24. Relé activado por 5Vcd [28] .....	31
Figura 25. Partes del relé. ....	32
Figura 26. Contactor trifásico 220v.....	34
Figura 27. Modulo bluetooth HC-05 [7]. .....	35
Figura 28. Diagrama de flujo en App inventor. ....	36
Figura 29. Diagrama de flujo de Arduino. ....	36
Figura 30. Vista de componentes de la plataforma en App inventor [7]. .....	37
Figura 31. Código desarrollado para el ingreso a la aplicación [7]. .....	38
Figura 32. Menú de pantalla principal de la AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN. ....	38
Figura 33. Código del menú principal de la aplicación Android. ....	40
Figura 34. Diagrama de flujo del código de Arduino [7] .....	55
Figura 35. Pantalla principal de la aplicación en dispositivo Android [7]. .....	56
Figura 36. Pantalla del menú para la AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN [7]. .....	56
Figura 37. Diagrama de flujo del código realizado para la aplicación Android [7]. .....	58
Figura 38. Conexión del sensor de temperatura. ....	59
Figura 39. Conexión del sensor ultrasónico.....	59
Figura 40. Conexión de relevador. ....	60
Figura 41. Salidas digitales para el control de sensores y actuadores.....	60
Figura 42. Conexión del módulo bluetooth .....	61
Figura 43. AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN [7]. .....	62

# Lista de tablas

<i>Tabla 1. Tiempo de cada proceso de la nixtamalización [7]</i> .....	10
<i>Tabla 2. Temperaturas del proceso de nixtamalización [7].</i> .....	12
<i>Tabla 3. pH del agua nixtamalizada [7]</i> .....	14
<i>Tabla 4. Adecuación de los actuadores y sensores para automatización [7].</i> .....	17

# Glosario de términos

<b>Palabras</b>	<b>Significado</b>
Cariópside	Fruto seco que tiene una sola semilla con el pericarpio adherido a la misma.
Consuetudinario	Que está basado en la costumbre o tradición.
Endospermo	Tejido del embrión de las plantas que le sirve de alimento.
Hemicelulosa	Polisacárido compuesto por más de un tipo de monómero).
Monómero	Molécula simple.
Nejayote	Caldo alcalino.
Nextli	cenizas de cal.
Pectina	Sustancia neutra.
Pecuaría	Del ganado o de la ganadería.
Pelagra	Enfermedad causada por la falta de ciertas vitaminas.
Pericarpio	Parte exterior del fruto de las plantas que envuelve las semillas.
Polisacárido	Hidrato de carbono, formado mediante la unión de varias moléculas de azúcar, como el almidón o la celulosa, que tienen una función estructural o energética (de reserva de glucosa).
Tamalli	Masa de maíz cocido.
timings	Tiempos.
triada	Conjunto de tres cosas.
embebidos	Es un subsistema electrónico diseñado para realizar una o algunas pocas funciones en tiempo real.

# Capítulo 1 Introducción

## 1.1. Antecedentes

La **nixtamalización** es el proceso mediante el cual se realiza la cocción del maíz con agua y cal, para obtener el nixtamal que, después de molido da origen a la masa nixtamalizada utilizada para la elaboración de **tortillas, tamales**, etc. Existen muchas evidencias de que este proceso se originó en Mesoamérica (específicamente en el altiplano mexicano). La palabra **nixtamal** proviene del **náhuatl** nextli (“cenizas de cal”) y tamalli (“masa de maíz cocido”), éste preparado tiene muchos usos, algunos de origen contemporáneo y otros de origen histórico [1]. Sin embargo poco se habla del proceso de nixtamalización que le confiere un alto valor nutritivo y cambios funcionales extraordinarios, y que es clave en la elaboración de la tortilla, el principal alimento en la dieta del pueblo mexicano y base de su supervivencia desde hace más de 3,500 años. [2]



*Figura 1. Maíz puro y maíz nixtamalizado [1]*

El maíz se encuentra en forma de mazorca; el grano como se muestra en la Figura 1 es una cariósida de forma aplastada. Su alto contenido en carbohidratos y proteínas lo hace el cereal ideal para todos los días. Se utiliza para la elaboración de jarabe y almidón; este último tiene aportes energéticos importantes para los seres humanos y es un proveedor de materias primas para la industria alimenticia, tanto humana como pecuaria. [3]

La semilla permite que se siembre tanto en suelos bien preparados como en suelos con mínima labranza. La siembra se puede realizar de forma manual depositando la semilla en los surcos o con maquinaria a una profundidad de 5 cm con una separación entre hileras de 60 a 80 cm y una separación entre plantas de 25 a 50 cm. [3]

El maíz es el cultivo más representativo de México por su importancia económica, social y cultural. Con un consumo promedio al año de 196.4 mil kg de maíz blanco, especialmente en tortillas, representa 20.9% del gasto total en Alimentos, Bebidas y Tabaco realizado por las familias mexicanas. [3]

## 1.2. Proceso de nixtamalización

Todos los pueblos alrededor del mundo tienen un grano de alimento base, que les aporta los nutrientes básicos que al complementarse con los demás alimentos de la región, han dado lugar a sistemas de agricultura inteligentes (como la milpa) y dietas saludables.

En el caso de México, este grano es el maíz. Si una población se alimenta de maíz en su estado original, se sufriría eventualmente de desnutrición y se padecerían enfermedades graves como la **pelagra**, debido a que no responde de las necesidades nutritivas del organismo humano. Sin embargo, los antiguos mexicanos encontraron una manera extraordinaria de resolver estas deficiencias, al cocerlo en agua alcalina, proceso conocido como nixtamalización. Sin el proceso de nixtamalización, es muy probable que los pueblos de Mesoamérica hubieran tenido que recurrir a otros productos, cuyo valor nutritivo se pudiera aprovechar mejor y la importancia del maíz hubiera sido menor, no hubiera sido la base de su alimentación. [4]

Aún hoy sorprende a la ciencia la invención del proceso de nixtamalización (o nixtamal), con el cual el maíz puede hacerse masa y se potencian sorprendentemente sus propiedades nutricionales.

Algunos estudios apuntan a que, en **Tenochtitlán**, miles de personas vivían bien nutridas solo con una dieta a base de maíz y nopal. ¿Por qué? Sucede que los alimentos mesoamericanos por excelencia son altamente nutritivos, y de hecho la mezcla de frijoles con maíz es altamente benéfica. Las tierras mesoamericanas, así, con la **triada** por excelencia, maíz, frijol, calabaza (y nopal y chile) habían proveído a los antiguos mexicanos de lo necesario para subsistir saludablemente.

Sin embargo, que fuese tan completa su dieta se debió también a **uno de los inventos más ingeniosos del hombre mesoamericano, la nixtamalización (o nixtamal)**; el proceso por el cual es posible hacer masa y con ella tortillas e infinidad de formas del maíz: tamales, sopes, tostadas, etc. [5].

## 1.3. Objetivo general

El objetivo se centra en la realización de un prototipo para llevar a cabo la automatización de una caldera para nixtamalización con interfaz Android.

Esta tesis tiene como objetivo el realizar un prototipo para la automatización de una caldera de nixtamalización para una tortillería; mediante una aplicación Android utilizada como interfaz de usuario para el control y monitoreo. La idea de este proyecto es realizar todo este procedimiento sin la continua verificación de un operador y se obtenga la misma cocción del maíz nixtamalizado.

**Por lo que el proyecto se planteó en los siguientes objetivos:**

1. Calcular y obtener la forma más adecuada para realizar el prototipo, eligiendo los componentes apropiados a las necesidades requeridas.
2. Realizar mediciones de diversas variables y periodos de tiempo para el diseño e implementación del código del prototipo
3. Comprobar el funcionamiento del prototipo diseñado, realizando mediciones y pruebas.

## 1.4. Justificación

La presente tesis se enfoca en realizar un prototipo de control, desarrollado en la plataforma de **Arduino Uno** y una aplicación en **App inventor**, para la automatización para caldera de nixtamal para beneficio de una tortillería.

Este prototipo automatiza el proceso de nixtamalización, y contribuye a la solución de un problema, basado en el tiempo invertido en realizar el proceso manualmente.

La idea de este prototipo es realizar el proceso de nixtamalización, sin la continua verificación de un operador y obtener la misma cocción del maíz; como consecuencia la calidad de la masa y en la producción de las tortillas se mejoró y se ahorró tiempo y dinero.

## 1.5. Metodología

Para la realización de esta tesis principalmente se hicieron:

- Pruebas de temperatura del agua.
- Tomas del tiempo que dura en alcanzar el punto óptimo de ebullición para que posteriormente se agregue la cal y maíz; y así llevar a cabo el proceso de nixtamalización.

Una vez obtenido estos datos se realizó:

- La aplicación para celular y el código, el cual nos facilitara el trabajo.

Por último:

- Implementación del prototipo.
- Comprobar su correcto funcionamiento.

## 1.6. Contenido de la Tesis

La tesis se desarrolló de la siguiente manera:

En el **capítulo 1** se detalla información de los antecedentes acerca del proceso de nixtamalización del maíz, fechas y descendencias de este proceso para sus diferentes usos.

En el **capítulo 2** se describe el estudio realizado para el proceso de nixtamalización.

En el **capítulo 3** se describe los sensores y actuadores a utilizar, y algoritmos para el proceso de nixtamalización mediante Arduino y la aplicación.

En el **capítulo 4** se detallan cada uno de los componentes, placa de Arduino y aplicación para celular.

En el **capítulo 5** se describe el funcionamiento del prototipo por diagrama de flujos

En el **capítulo 6** se presentan las pruebas y funcionamiento del prototipo

En el **capítulo 7** se describen conclusiones.

# Capítulo 2 Estudio del proceso de la Nixtamalización

## 2.1. Estudio general de nixtamalización y planteamiento del algoritmo.

Para entender mejor el proceso que se lleva a cabo se realizó un diagrama de flujo como el de la Figura 2.



Figura 2. Diagrama de flujo del proceso de nixtamalización

El primer paso en la nixtamalización consiste en llenar la caldera de agua, el paso siguiente es calentar el agua a una temperatura cercana al punto de ebullición, colocar los granos de maíz en una solución alcalina (agua más hidróxido de calcio o cal). La cocción y remojo varían según el tipo de maíz, las tradiciones locales y el tipo de alimentos a preparar. Se puede cocer desde unos minutos hasta unas horas, y dejar remojando desde unos minutos hasta alrededor de un día.

Durante la cocción y el remojo, tienen lugar una serie de cambios químicos en los granos de maíz, debido a que los componentes de la membrana celular de los granos, entre los cuales se incluyen hemicelulosa y pectina, son altamente solubles.

En soluciones alcalinas, los granos:

- Se suavizan.
- Su pericarpio se afloja.
- El grano se hidrata y absorbe calcio y potasio (según los compuestos utilizados) a lo largo de todo el proceso.
- Los almidones se disuelven y gelatinizan, y algunos se dispersan en el líquido.
- Se liberan ciertos productos químicos del germen Figura 3 que permiten que el grano cocido sea más fácil de triturar.
- La cocción produce cambios en la proteína principal del maíz, **lo que hace que las proteínas y nutrientes del endospermo** Figura 3 **en su núcleo sean más asimilables para el cuerpo humano** [1].
- El incremento en el balance de los aminoácidos esenciales.



Figura 3. Partes que conforman el maíz [6].

Para que la nixtamalización cumpla mejor su objetivo es necesario monitorear o medir su pH, ya que un valor de pH alto se puede obtener al incrementar la cantidad de cal durante el cocimiento, o al reducir el lavado del nixtamal.

Después el grano se usa solo o se muele para poder obtener harina de maíz. Este proceso se ha modernizado y actualmente la molienda se lleva a cabo mayoritariamente con máquinas o **molinos industriales** Figura 4, pero en las áreas rurales persiste el uso del metate como se muestra en la Figura 5, que es un molino prehispánico de piedra volcánica [1].

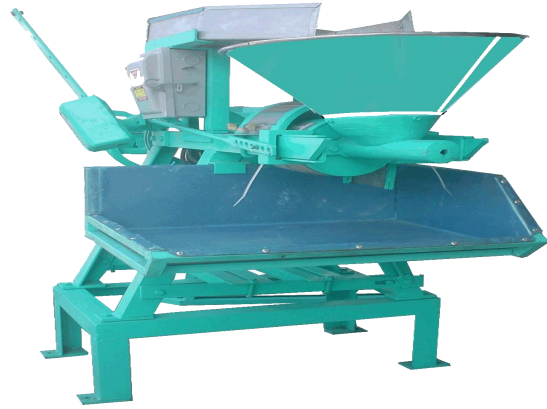


Figura 4. Molino industrial [4]



Figura 5. Molino de metate [1]

## 2.2. Estudio de tiempo del proceso de nixtamalización

El proceso que realizamos para la nixtamalización, se lleva a cabo principalmente con el llenado de la caldera, con agua hasta lo requerido, dependiendo de la cantidad de maíz que se quiera cocer. Lo siguiente una vez teniendo el agua a nivel, se calienta el agua hasta llegar al punto óptimo de ebullición, estando en su punto óptimo de temperatura se le agrega la cantidad adecuada de cal, esto; cuando se tiene el agua hirviendo con cal, se le agrega la cantidad de maíz y se deja en reposo por más de 12 horas o todo un día; después del tiempo de reposo se puede utilizar.

Se dice que está listo cuando se agarra un puño de nixtamal y se frota sobre la mano, y si este se desprende sin esfuerzo nos indica que está en su punto óptimo; por lo contrario si agarramos y frotamos y es difícil para desprender, esto quiere decir que requiere algunos minutos más en el proceso de reposo para su cocimiento. El nixtamal debe estar frío para poder darle la textura deseada que se necesita para el uso que se requiera.

Algunas personas prefieren el nixtamal no bien cocido, esto por el hecho de que al no despellejarse fácilmente, a la hora de enjuagarlo para retirar el nejayote del nixtamal, no se despelleje tanto y contenga una de sus capas que es el pericarpio, esto con el fin de que contenga una textura pegajosa para el uso final que se le requiera dar sea mejor y no quede muy reseco el producto final.

Para lo cual se realizó un estudio general de la nixtamalización y se obtuvieron estos tiempos de cada paso del proceso.

#### **Tiempo general de cada proceso para la elaboración del nixtamal:**

1. Se midió el tiempo que tarda en llenar la caldera de agua (**el tiempo aproximado es de 20 a 45 min**).
2. Se midió el tiempo que tarda en llegar a su punto óptimo de cocimiento (**el tiempo aproximado es de 20 a 30 min**).
3. Posteriormente pasado ese tiempo, se realizó la acción de agregar la cal en el agua en su punto de ebullición.
4. Una vez teniendo el agua con cal, se sube el maíz por un ducto para vaciarlo en la caldera con el agua y cal ya mezclados (**el tiempo aproximado es de 2 a 10 min**).

En la Figura 6 se muestra la caldera que se automatizo y las partes que la conforman, como la llave de agua, de gas, motor para subir el maíz a la caldera. Estas se cambiaron por partes eléctricas o electrónicas de control, así adecuamos la caldera para su correcta automatización.

Se llevó a cabo un estudio diario, durante una semana y se midió el tiempo que tarda el proceso de la nixtamalización, para ello se realizó una tabla de los tiempos de cada una de los etapas del proceso de nixtamalización como se muestra en la Tabla 1.

Para la nixtamalización el proceso varia, ya que no se cuenta con un método que nos proporcione el tiempo de cada paso del proceso, por ello se tomó la decisión de hacer un prototipo que pueda realizar todo este proceso con la mínima supervisión de un operador, haciendo que el tiempo de cada paso del proceso esté controlado por un **Arduino Uno**, el cual nos indica cuando esté listo el proceso de nixtamalización.



Figura 6. Caldera para la nixtamalización [7].

Tabla 1. Tiempo de cada proceso de la nixtamalización [7]

Etapa del Proceso	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Cantidad de maíz	300KG	350KG	350KG	350 KG	400 KG	400 KG	350 KG
Llenado de agua	25 min	25 min	24 min	23 min	27 min	28 min	23 min
Punto de ebullición	24 min	25 min	27 min	33 min	35 min	40 min	28 min
Vaciado de maíz	10 min	11 min	10 min	11 min	12 min	12 min	10 min

### LLENADO DE AGUA PARA LA CALDERA

En esta parte del proceso es donde se inicia el proceso, la cantidad de agua esta delimitada por la cantidad de maíz que se requiera nixtamalizar, como podemos observar en la Figura 7. Por ejemplo, si la cantidad de agua es para 400kg de maíz, esta tarda más en ser calentada y por consiguiente al agregar una cantidad menor de maíz para la cantidad de agua que fue puesta en la caldera; como por ejemplo 250kg, este no podrá bajar la temperatura del agua y al momento de que el nixtamal se encuentre en reposo, tendrá más temperatura el agua

y cocerá de más el maíz; en cambio si se pone una cantidad adecuada a la cantidad de maíz por cocer, este podrá bajar la temperatura del agua y al estar en reposo se cocerá con la temperatura adecuada sin ningún problema de que este se pase de cocimiento.



*Figura 7. Paso número 1 del proceso para la nixtamalización [7].*

### **CALENTAMIENTO DEL AGUA EN LA CALDERA**

En la Figura 8 se muestra el proceso encargado de calentar el agua a una temperatura cercana al punto de ebullición para posteriormente agregar la cal y el maíz, el agua tiene su pH en la zona neutra con 7 en la escala, pero al momento de estar expuesta a la temperatura, el agua se va alcalinizando hasta alcanzar un 8 o 9 de la escala del pH, pero, esto no sería suficiente para alcalinizar el agua adecuadamente, para que el maíz logre que el pericarpio se afloje y entre en el endospermo y el germen de los granos de maíz, es necesario alcanzar una alcalinidad alta, se le agrega la cal que es llamado a su compuesto **hidróxido de calcio**, lo que hace al reaccionar con el agua caliente es tener una alcalinidad con un pH de 12 o mayor dependiendo de qué tan alcalino queremos que sea nuestro nejayote.



Figura 8. Paso número 2 del proceso para la nixtamalización [7].

## 2.3. Estudio de temperatura del proceso de nixtamalización

El estudio realizado de las temperaturas sobre el proceso de nixtamalización. Se llevó a cabo durante una semana, para tener algunas referencias de cómo cambia la temperatura, en relación a los otros días con las mismas cantidades de maíz como se muestra en la Tabla 2. Esto con la finalidad de hacer un estudio y tener un mejor rango de temperatura y la calidad del nixtamal sea siempre la misma.

Tabla 2. Temperaturas del proceso de nixtamalización [7].

Proceso	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Llenado de caldera con agua	45 °C	47 °C	46 °C	44 °C	46 °C	59 °C	47 °C
Empezando a ebullición	86 °C	88 °C	88 °C	89 °C	88 °C	88 °C	91 °C
Punto de ebullición	90 °C	92 °C	91 °C	91 °C	94 °C	95 °C	95 °C
Con Cal	90 °C	92 °C	91 °C	91 °C	94 °C	95 °C	95 °C

Una vez teniendo el agua a la temperatura requerida se procede a agregar la cantidad de cal en la caldera como se puede observar en la Figura 9.



*Figura 9. Paso número 3 del proceso para la nixtamalización [7].*

Como se muestra en la Figura 10. Teniendo el agua con la cal disuelta en la caldera, se procede a apagar la flama del gas y se agrega el maíz requerido para nixtamalizar.



*Figura 10. Paso número 4 del proceso para la nixtamalización [7].*

## 2.4. Estudio de pH del agua nixtamalizada

El pH es una medida utilizada en la Química, se usa para medir el grado de acidez o alcalinidad de una sustancia, por lo general en su estado líquido. Por lo tanto, el pH mide la concentración de iones de hidrógeno de una sustancia. Esto se mide en una escala del 0 al 14 donde el cero es Acido y el 7 se considera como Neutro, el número 14 es Alcalino.

El estudio que se realizó sobre el agua del nixtamal, (Nejayote) fue para conocer el grado de pH en el proceso de nixtamalización, esto se estuvo midiendo durante una semana los resultados se muestran en la Tabla 3.

El análisis del nejayote se mandó para ser analizado con un equipo como el de la Figura 11 a un **centro de investigación biomédica de Michoacán (IMSS)**.



Figura 11. Equipo medidor de pH [8]

Tabla 3. pH del agua nixtamalizada [7]

Día	pH
Lunes	6.42
Martes	11.38
Miércoles	11.58
Jueves	11.44
Viernes	11.50
Sábado	11.56
Domingo	11.53

## 2.5. Maíz en reposo y nixtamalizado

Este es el último proceso, el cual hace que los granos se suavicen y sus pericarpios se aflojen. El grano se hidrata y absorbe calcio y potasio (según los compuestos utilizados) a lo largo de todo el proceso. Los almidones se disuelven y gelatinizan, y algunos se dispersan en el líquido. Se liberan ciertos productos químicos del germen que permiten que el grano cocido sea más fácil de triturar. La cocción produce cambios en la proteína principal del maíz, lo que hace que las proteínas y nutrientes del endospermo del núcleo sean más asimilables para el cuerpo humano [1].



*Figura 12. Maíz recién Nixtamalizado [7].*



*Figura 13. Maíz Nixtamalizado en proceso de reposo [7].*

# Capítulo 3 Automatización del proceso de nixtamalización

## 3.1. Introducción

El proceso de nixtamalización es de origen prehispánico, se originó en Mesoamérica y tiene más de 3,500 años alimentando al pueblo mexicano. Es una de las bases alimenticias y el 20% del gasto diario de los mexicanos. Si queremos consumir maíz es necesario que esté nixtamalizado, de otra forma no aportaría nutrientes y no sería comestible. Para cualquier tipo de uso que se le quiera dar al maíz ya sea para tortilla, masa, pozole, tamales, etc. Siempre estará presente el proceso de nixtamalización, ya que este proceso hace que el grano de maíz absorba nutrientes como lo es el calcio y potasio, el germen mediante el cambio químico que ocurre al estar en una agua alcalina permite que sea más fácil de triturar. La nixtamalización es el contacto del maíz, el agua, el calor y el hidróxido de calcio.

Este proceso de nixtamalización se hace mediante un personal dedicado específicamente a este proceso, por lo cual se tomó la decisión de hacer un prototipo que pueda automatizar el proceso aumentando la calidad y cuidando el proceso de nixtamalización.

Esta automatización es controlada por una placa electrónica denominada **Arduino Uno** en base a un microcontrolador ATMEGA328, la cual funciona como el centro del proceso de nixtamalización. Para que este microcontrolador funcione es necesario que el operador proporcione la orden de iniciar, por ello se desarrolla una aplicación en la plataforma **App Inventor**, el cual es una plataforma de aplicaciones en un lenguaje de programación a bloques, es un tanto más fácil realizar un programa de esta forma solo necesitas tener conocimientos básicos de programación. La aplicación desarrollada será la encargada de hacer la interface entre el operador y el controlador de la nixtamalización, el control cuenta con sensores y actuadores encargados de obtener datos y enviar datos al **Arduino Uno** para que tome decisiones con los actuadores.

## 3.2. Sensores y Actuadores del sistema.

### Definición de Sensor.

Un sensor es un dispositivo que está capacitado para detectar acciones o estímulos externos y responder en consecuencia. Estos aparatos pueden transformar las magnitudes físicas o químicas en magnitudes eléctricas [9].



## Definición de Actuador.



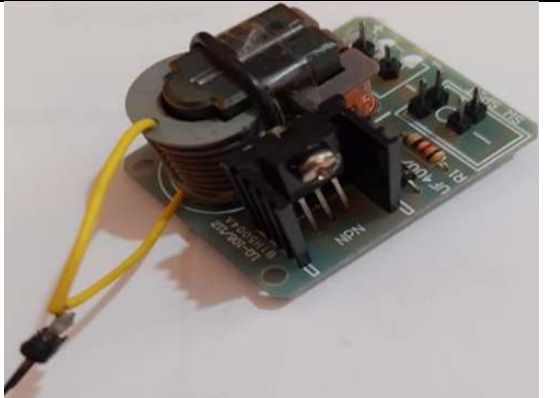
Es un dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o **“actuar”** otro dispositivo mecánico. La fuerza que provoca el actuador proviene de tres fuentes posibles: Presión neumática, presión hidráulica, y fuerza motriz eléctrica (motor eléctrico o solenoide). Dependiendo del origen de la fuerza, el actuador se denomina **“neumático”**, **“hidráulico”** o **“eléctrico”**.

## Historia

El actuador más común es el actuador manual o humano. Es decir, una persona mueve o actúa un dispositivo para promover su funcionamiento. Con el tiempo, se hizo conveniente automatizar la actuación de dispositivos, por lo que diferentes dispositivos hicieron su aparición. Actualmente hay básicamente dos tipos de actuadores. Lineales y Rotatorios [10]

*Tabla 4. Adecuación de los actuadores y sensores para automatización [7].*

<ul style="list-style-type: none"><li>• <b>Para el prototipo se cambió la llave de agua por una electroválvula para agua la cual estará controlada por el sistema de control.</b></li></ul>	
<ul style="list-style-type: none"><li>• <b>La llave de gas se reemplazó por una electroválvula para Gas LP la cual estará controlada por el sistema de control.</b></li></ul>	

<ul style="list-style-type: none"> <li>• El motor será encendido mediante un contactor el cual estará controlado por un relevador que se activara por el sistema de control.</li> </ul>	
<ul style="list-style-type: none"> <li>• El sensor de temperatura estará indicando la temperatura, señal que el sistema de control utiliza para determinar la temperatura requerida.</li> <li>• El sensor ultrasónico nos indicara el nivel de agua, señal que el sistema de control utiliza para determinar el nivel requerido.</li> </ul>	
<ul style="list-style-type: none"> <li>• El generador de chispa permite iniciar la quema del gas proporcionada por la electroválvula que controla el flujo de gas LP.</li> </ul>	

### 3.3. El Arduino Uno como elemento de control

El **Arduino Uno** es una placa electrónica muy económica y fácil de conseguir, contiene entradas/salidas digitales y entradas analógicas necesarias para el prototipo, como también es fácil de cargar el código necesario a la placa, no necesitamos algún otro programa, más que el mismo programa de **Arduino Uno** donde se realiza la programación.

#### Características de Arduino Uno

- Microcontrolador: **ATmega328**.
- Voltaje de operación: **5V**.

- Voltaje de entrada (recomendado): **7-12V**.
- Voltaje de entrada (límites): **6-20V**.
- Pines de E/S digitales: **14 (de los cuales 6 proporcionan salida PWM)**.
- Pines de entrada analógica: **6**.
- Corriente DC por pin de E/S: **40 mA**.
- Memoria Flash: **32 KB de los cuales 0,5 KB utilizados por el bootloader**.
- SRAM: **2 KB (ATmega328)**.
- EEPROM: **1 KB (ATmega328)**.
- Velocidad de reloj: **16 MHz**.

El **Arduino Uno** se diferencia de todas en que no utiliza el FTDI USB a serie driver chip. En lugar de ello, cuenta con el Atmega8U2 programado como convertidor de USB a serie [11].

### 3.3.1. Hardware

Es una palabra de origen inglés con la que se hace referencia a toda la parte dura de la informática, es decir a la maquinaria real utilizada para el procesamiento electrónico de datos [12].

Corresponde a todas las partes tangibles de un sistema informático. Sus componentes son: eléctricos, electrónicos, electromecánicos, microprocesadores y mecánicos son hardware [13].

La parte del hardware estará constituida por:

1. Sensores.
2. Actuadores
3. Indicadores

### 3.3.2. Software

Conjunto de programas, instrucciones, datos y reglas informáticas para ejecutar ciertas tareas en una computadora. [14].

El programador es la persona que utiliza software de desarrollo para crear programas que satisfagan necesidades específicas.

**Software de programación:**

Es el conjunto de herramientas que permiten al programador desarrollar programas informáticos, usando diferentes alternativas y lenguajes de programación, de una manera práctica. Incluyen básicamente:

- **Editores de texto:** son programas que permiten crear y modificar archivos digitales compuestos únicamente por texto sin formato, conocidos comúnmente como archivos de texto o texto plano. El programa lee el archivo e interpreta los bytes leídos según el código de caracteres que usa el editor. Hoy en día es comúnmente de 7 u 8-bits en ASCII [15].
- **Compiladores:** reúne diversos elementos o fragmentos en una misma unidad. En el terreno de la informática, compilar consiste en traducir un programa escrito en un cierto lenguaje a otro [16].
- **Intérpretes:** es un software que recibe un programa en lenguaje de alto nivel, lo analiza y lo ejecuta. Para analizar el programa completo, va traduciendo sentencias de código y ejecutándolas si están bien, así hasta completar el programa origen [17].
- **Enlazadores:** es un programa que toma los objetos generados en los primeros pasos del proceso de compilación, la información de todos los recursos necesarios (biblioteca), quita aquellos recursos que no necesita, y enlaza el código objeto con su(s) biblioteca(s) con lo que finalmente produce un fichero ejecutable o una biblioteca [18].
- **Depuradores:** es una herramienta de software que permite ejecutar un programa, instrucción a instrucción, para comprobar su correcto funcionamiento [19].
- **Entornos de Desarrollo Integrados (IDE):** Agrupan las anteriores herramientas, usualmente en un entorno visual, de forma tal que el programador no necesite introducir múltiples comandos para compilar, interpretar, depurar, etc. [14].

**Cuando una persona o empresa no encuentra en el mercado programas que satisfagan sus necesidades, encarga a un programador el desarrollo de software personalizado a sus propios intereses** [14]. Es por ello que se requirió hacer la AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN, el cual contará de un entorno de desarrollo integrado como lo que es el programa Arduino IDE con la placa de **Arduino Uno**, este microcontrolador se programó con todas las funciones a realizar, para que el proceso de nixtamalización se lleve a cabo.

De la misma manera, se realizó un programa pero para una aplicación celular, con un lenguaje de programación a bloques, cuenta con instrucciones, datos y reglas a utilizar. En su diagrama de flujo siempre se encuentra esperando un evento para realizar una acción, en la aplicación desarrollada siempre estará esperando a que el usuario elija una cantidad de bultos por nixtamalizar, para que realice la acción de mandar el dato al Arduino y se ejecute ese proceso.

### 3.4. Aplicación en el celular para control del sistema

Para controlar la nixtamalización de forma automática, se desarrolló una aplicación en **App inventor**; la cual es muy fácil y práctica para el usuario. Cuenta con una pantalla principal el cual nos pide ingresar un usuario y una contraseña para poder acceder al menú de la automatización de la caldera, estando en el menú cuenta con botones, para la cantidad de maíz que se quiera nixtamalizar y botones para la conexión vía bluetooth, bloqueo de botones para no presionar accidentalmente otro, salida de la misma aplicación.

La aplicación será la interfaz entre el usuario y el control con **Arduino Uno**.

### 3.5. Algoritmo de control

El algoritmo usado es secuencial y realiza las siguientes acciones:

1. Colocar el Maíz en su depósito.
2. Si hay maíz en el depósito se abre la electroválvula del agua en la caldera.
3. Se espera a que se llene la caldera de agua.
4. Se abre la electroválvula del gas LP y se inicia su ignición, para calentar el agua
5. Se espera a que el agua este a la temperatura adecuada
6. Se cierra la electroválvula del gas LP y se le agrega cal a la caldera.
7. Se sube el maíz del depósito de maíz a la caldera.
8. Cuando se tiene todo el maíz en la caldera con agua y cal, se vacía el contenido de la caldera en una tina de reposo, para que se realice la nixtamalización.

# Capítulo 4 Sensores y actuadores del sistema a controlar

## 4.1. Introducción

Los sensores, hoy en día se han vuelto una parte indispensable en la vida industrial, ya que estos permiten detectar las variaciones de variables físicas.

Existen en todas las empresas, casas, automóvil ya que son usados en forma rutinaria, haciendo que el sistema disponga de la información necesaria de su entorno. Un sensor es un transductor que convierte algún fenómeno físico en señales eléctricas que el microprocesador puede leer, de igual forma los actuadores que son capaces de generar una fuerza, a partir de líquidos y energía eléctrica.

Para que un sistema electrónico de control pueda controlar un proceso o producto es necesario que pueda actuar sobre el mismo.

## 4.2. Arduino Uno

Esencialmente la placa **Arduino Uno** es una placa electrónica basada en el microcontrolador ATmega328.

Para programar la board se necesita el *IDE Arduino*.

Tiene:

- **Voltaje Operativo:** 5v.
- **Voltaje de Entrada (Recomendado):** 7 – 12 v.
- **Pines de Entradas/Salidas Digital:** 14 (De las cuales 6 son salidas PWM).
- **Pines de Entradas Análogas:** 6.
- **Memoria Flash:** 32 KB (ATmega328) de los cuales 0,5 KB es usado por Bootloader.
- **SRAM:** 2 KB (ATmega328).
- **EEPROM:** 1 KB (ATmega328).
- **Velocidad del Reloj:** 16 MHZ.

El software de la placa incluye un controlador USB que puede simular un ratón, un teclado y el puerto serie.

Una vista superior de la placa se muestra en la Figura 14, este es el esquema de lo que puedes ver (los componentes de la placa con los que puedes interactuar en su uso normal están resaltados) [20]:

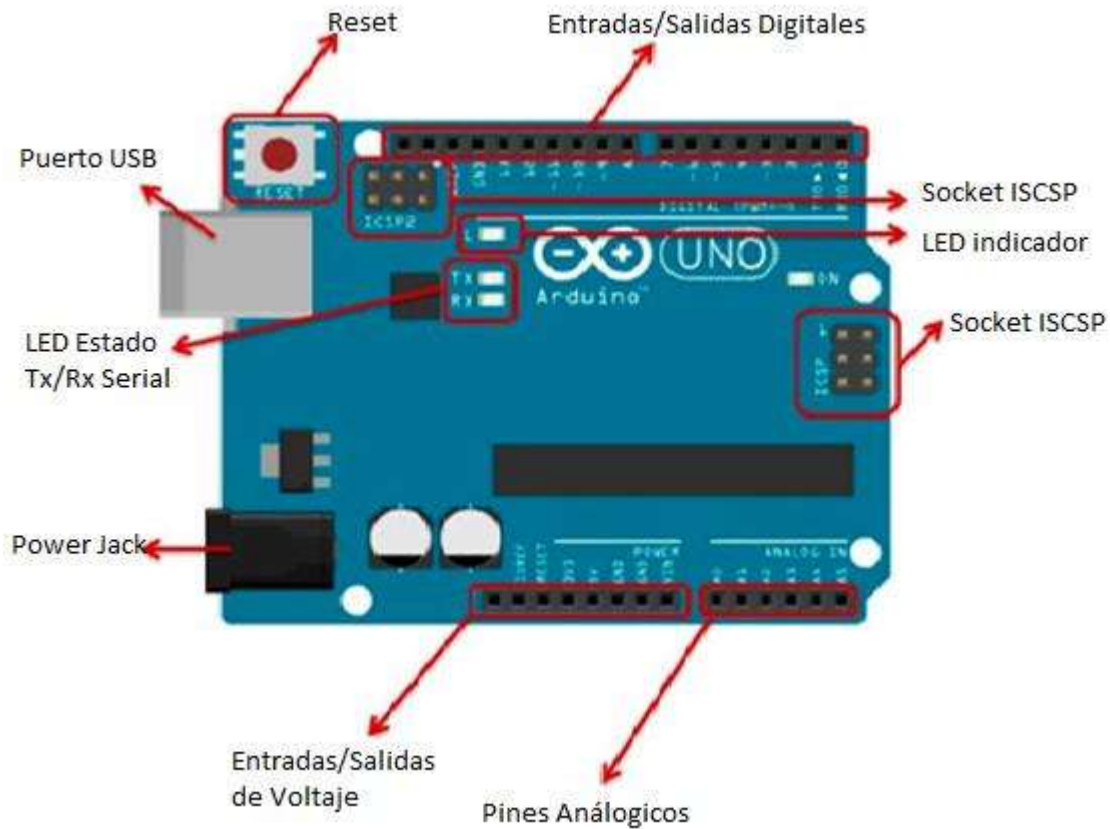


Figura 14. Arduino Uno descripción de componentes [21]

### Empezando según las agujas del reloj:

- Terminal de referencia analógica.
- Tierra digital.
- Terminales digitales 2-13.
- Terminales digitales 0-1/ E/S serie. TX/RX. Estos pines no se pueden utilizar como e/s digitales (`digitalRead()` y `digitalWrite()`) si estás utilizando comunicación serie (por ejemplo `Serial.begin()`).
- Botón de reinicio (reset).
- Programador serie en circuito (In-circuit Serial Programmer) o (ICSP).
- Terminales de entrada analógica 0-5.
- Terminales de alimentación y tierra.
- Entrada de alimentación externa (9-12VDC, power Jack).

- Puerto USB (utilizado para subir programas a la placa y para comunicaciones serie entre la placa y el ordenador; puede utilizarse como alimentación de la placa) [20].

### **Entradas y salidas:**

Cada uno de los 14 pines digitales se puede usar como entrada o como salida. Funcionan a 5V, cada pin puede suministrar hasta 40 mA. La intensidad máxima de entrada también es de 40 mA.

Cada uno de los pines digitales dispone de una resistencia de pull-up interna de entre 20K $\Omega$  y 50 K $\Omega$  que está desconectada, salvo que nosotros indiquemos lo contrario.

Arduino también dispone de 6 pines de entrada analógicos que trasladan las señales a un conversor analógico/digital de 10 bits.

### **Pines especiales de entrada y salida:**

- **RX y TX:** Se usan para transmisiones serie de señales TTL.
- **Interrupciones externas:** Los pines 2 y 3 están configurados para generar una interrupción en el atmega. Las interrupciones pueden dispararse cuando se encuentra un valor bajo en estas entradas y con flancos de subida o bajada de la entrada.
- **PWM:** Arduino dispone de 6 salidas destinadas a la generación de señales PWM de hasta 8 bits.
- **SPI:** Los pines 10, 11, 12 y 13 pueden utilizarse para llevar a cabo comunicaciones SPI, que permiten trasladar información full dúplex en un entorno Maestro/Esclavo.
- **I<sup>2</sup>C:** Permite establecer comunicaciones a través de un bus **I<sup>2</sup>C**. El bus **I<sup>2</sup>C** es un producto de Phillips para interconexión de sistemas **embebidos**. Actualmente se puede encontrar una gran diversidad de dispositivos que utilizan esta interfaz, desde pantallas LCD, memorias EEPROM, sensores...

### **Alimentación de un Arduino**

Puede alimentarse directamente a través del propio cable USB o mediante una fuente de alimentación externa, como puede ser un pequeño transformador o, por ejemplo una pila de 9V. Los límites están entre los 6 y los 12V. Como única restricción hay que saber que si la placa se alimenta con menos de 7V, la salida del regulador de tensión a 5V puede dar menos que este voltaje y si sobrepasamos los 12V, probablemente dañaremos la placa.

La alimentación puede conectarse mediante un conector de 2,1mm con el positivo en el centro o directamente a los pines Vin y GND marcados sobre la placa.

Hay que tener en cuenta que podemos medir el voltaje presente en el jack directamente desde Vin [22].

## 4.2.1. Componentes

A continuación se describen los componentes utilizados para controlar el sistema de nixtamalización.

### 4.2.1.1. Sensores

#### 4.2.1.1.1. Sensor de temperatura DS18B20

Está formado por un procesador con múltiples módulos Figura 15, que se encargan de controlar la comunicación, medir la temperatura, y gestionar el sistema de alarmas.

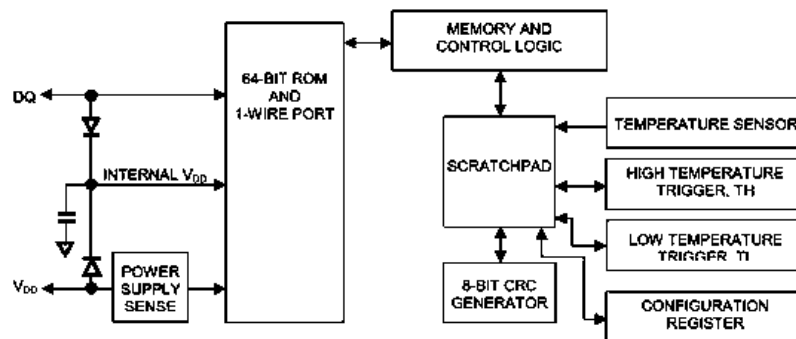


Figura 15. Diagrama de bloques del Termómetro digital DS18B20.

Una de las principales ventajas de DS18B20 es su bus de comunicación 1-Wire que le permite realizar la transmisión empleando únicamente un cable de datos. Para ello, 1-Wire está basado en un complejo sistema de **timings** en la señal, entre el dispositivo emisor y el receptor. La mayor desventaja del sistema 1-Wire es que requiere un código complejo, lo que a su vez supone una alta carga del procesador para consultar el estado de los sensores. El tiempo de adquisición total de una medición de 750ms [22].

La resolución del DS18B20 es configurable a 9, 10, 11 o 12 bits, siendo 12 bits el modo por defecto. Esto equivale a una resolución en temperatura de 0.5°C, 0.25°C, 0.125°C, o 0.0625°C [22], respectivamente. Algo muy importante es saber qué rango de temperaturas es capaz de medir un sensor de este tipo. No es lo mismo medir la temperatura ambiente de una casa en que medir la temperatura de un congelador.

El **DS18B20 puede medir temperaturas entre -55°C y 125°C**. Es un rango muy amplio sin embargo, no en todo el rango tenemos el mismo error [23]. Para temperaturas entre -10°C y 85°C podemos tener  $\pm 0.5^\circ\text{C}$ . Para el resto de temperaturas entre -55°C y 125°C el error es de  $\pm 2^\circ\text{C}$  como se muestra en la Figura 16.

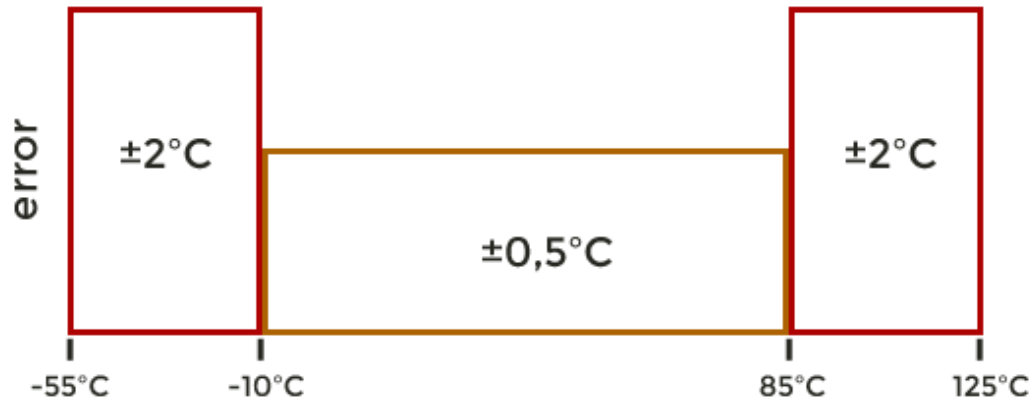


Figura 16. Gráfica de errores con relación a la temperatura [23].

El tipo de conexión de este sensor es del tipo clave 3 como se muestra en la Figura 17 donde:

- Cable rojo: V<sub>DD</sub>
- Cable amarillo: DQ
- Cable negro: GND



Figura 17. Termómetro DS18B20 [22].

#### 4.2.1.1.2. Sensor ultrasónico

El sensor HC-SR04 es un sensor ultrasónico que permite medir distancia, es de bajo costo y realiza mediciones de distancia de objetos en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio. Es el más utilizado dentro de los sensores de tipo ultrasónico, principalmente por la cantidad de información que hay de él. De igual forma es empleado en proyectos de robótica o de automatización como sistemas de medición de nivel o distancia.

Posee dos transductores: un emisor y un receptor piezoeléctricos, además de la electrónica necesaria para su operación.

El funcionamiento del sensor es el siguiente:

- El emisor piezoeléctrico emite 8 pulsos de ultrasonido (40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico como se muestra en la Figura 18, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECHO es medido por el microcontrolador y así se puede calcular la distancia al objeto. El funcionamiento del sensor no se ve afectado por la luz solar o material de color negro (aunque los materiales blandos acústicamente como tela o lana pueden llegar a ser difíciles de detectar) [24].

La distancia se puede calcular utilizando la siguiente fórmula:

$$\text{Distancia (m)} = \{(\text{Tiempo del pulso ECO}) * (\text{Velocidad del sonido}=340\text{m/s})\}/2$$

$$\text{Distancia (m)} = \frac{(T_{ECO} [\text{Seg}]) \times \left(C \left[\frac{\text{m}}{\text{Seg}}\right]\right)}{2}$$

Donde:

T\_ECO es el tiempo del pulso ECO en Segundos

C es la velocidad del sonido en  $\frac{\text{m}}{\text{Seg}}$

#### Rangos de medición efectiva:

El rango de medida del sensor va de 2cm a 400cm con una resolución de 0.3cm y su ángulo de detección es de 30° como en la Figura 19, aunque para medidas más eficaces se recomienda estar dentro de los 15° [25].

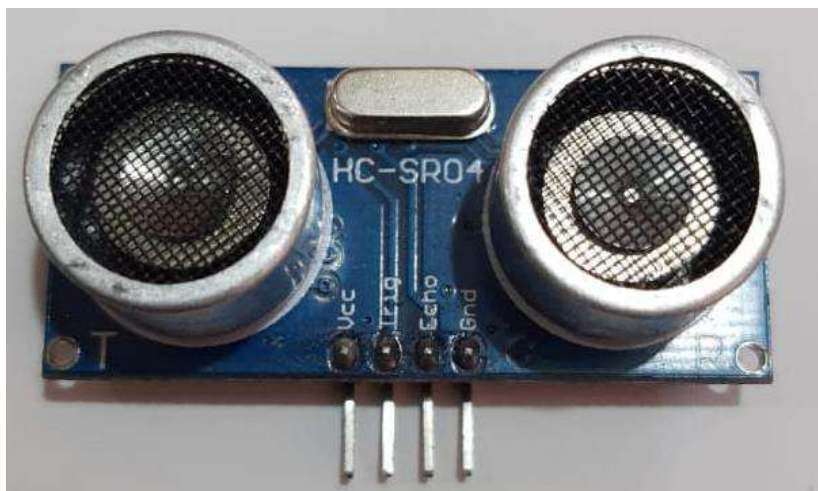


Figura 18. Sensor Ultrasónico [7].

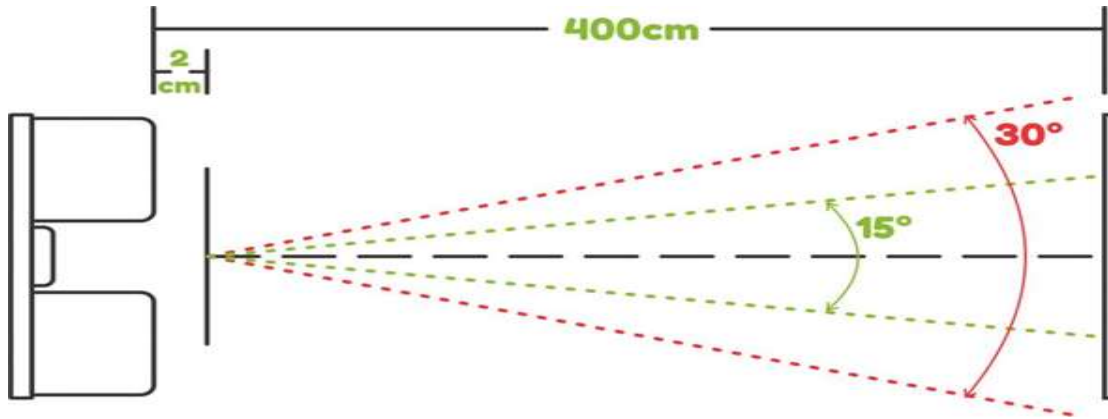


Figura 19. Rango de medición sensor HC-SR04 [25].

#### 4.2.1.2. Actuadores

##### 4.2.2.1.1. Válvula de Agua

Para controlar el flujo de un líquido usando micro controlador. La válvula solenoide de 12V. Solo se debe conectar a una fuente de fluido (agua) de 3/4" de entrada de rosca y se interrumpirá el flujo de agua hasta que la conecte a 12 Voltios. Los 12 Voltios se aplican a los conectores que vienen en la válvula de la Figura 20.



Figura 20. Electroválvula solenoide para agua [21].

Se probó con presiones de la red local de agua sin ningún inconveniente.

También puede ser alimentada desde un estanque bajo las siguientes condiciones:

- Altura del estanque debe ser de dos metros o más.
- Voltaje de funcionamiento: 11 a 12 voltios DC.
- Corriente de funcionamiento: 340 mA.
- Rosca externa de 3/4" (dimensión bastante popular en casas).
- Filtro de acero inoxidable para prevenir daños.
- Temperatura máxima del fluido 122 °C.
- Con este rango de presiones, esta válvula se puede utilizar en campos, predios e incluso en una red domiciliaria.

**NOTA IMPORTANTE:** Esta válvula posee cierre automático. Es decir, en caso de corte de la energía eléctrica no perderá líquido pues se cierra automáticamente [21].

#### 4.2.2.1.2. Válvula de Gas

Válvula Solenoide 8210G76

2-Vías normalmente cerrada, operación de control comercial e industrial de quemadores de gas, Ideal para aplicaciones de alta presión, cuerpo de latón construcción, montaje en cualquier posición ¾" NPT, 120VCA [26]

Características:

- Amplia gama de clasificaciones de presión, tamaños y resistencia
- Los materiales proporcionan una larga vida útil y baja fuga interna.
- Válvulas de alto flujo para líquidos, corrosivos y aire/inerte, servicio de gas.

Las aplicaciones industriales incluyen:

- Lavado de autos.
- Lavandería.
- Compresores de aire.
- Control de agua industrial.
- Bombas.



Figura 21. Válvula solenoide para Gas LP [26].

#### 4.2.1.3. Generador de chispa

Una **chispa eléctrica** es una descarga eléctrica repentina que ocurre cuando un campo eléctrico suficientemente alto crea un canal conductor ionizado eléctricamente en un medio normalmente aislante, que suele ser aire, otros gases o diversas mezclas de gases.

La transición rápida desde un medio no conductor a un estado conductor produce una breve emisión de luz y un sonido agudo o como un chasquido. Una chispa se crea cuando el campo eléctrico aplicado supera la rigidez dieléctrica del medio en el que se produce. Para aire, la rigidez dieléctrica es aproximadamente 30 kV/cm a nivel de mar [27].

Este generador de chispa de 15 Kv se activa por medio de un Relé Figura 24, este es activado por una señal digital del **Arduino Uno**, la cual al mismo tiempo que se abre la válvula de gas para generar una flama. Se puede observar en la Figura 22 la conexión del módulo generador de chispa y el modulo real en la Figura 23.

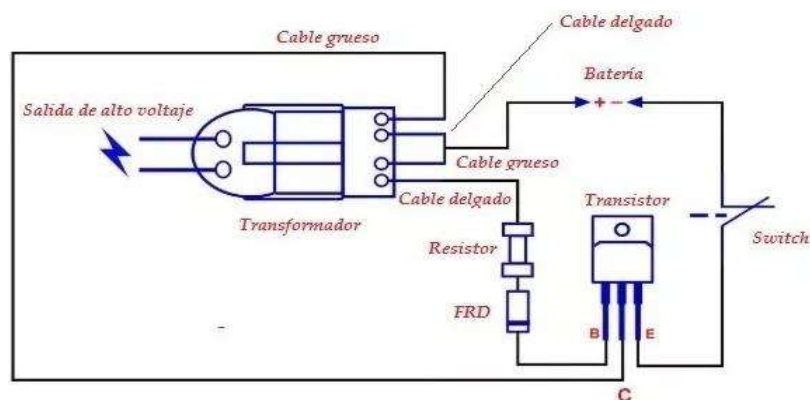


Figura 22. Diagrama de conexión del módulo generador de chispa.

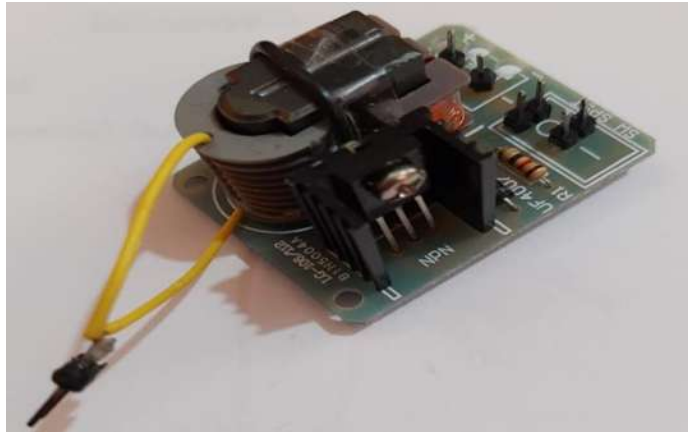


Figura 23. Modulo generador de chispa 15kv [7].



Figura 24. Relé activado por 5Vcd [28]

#### 4.2.1.4. Contactor para motor

##### Los relés

Su funcionamiento se basa en el fenómeno electromagnético. Cuando la corriente atraviesa la bobina, produce un campo magnético que magnetiza un núcleo de hierro dulce (ferrita) como se ilustra en la Figura 25. Este atrae al inducido que fuerza a los contactos a tocarse. Cuando la corriente se desconecta vuelven a separarse. Es un dispositivo que consta de dos circuitos diferentes: un circuito electromagnético (electroimán) y un circuito de contactos, al cual aplicaremos el circuito que queremos controlar [29]

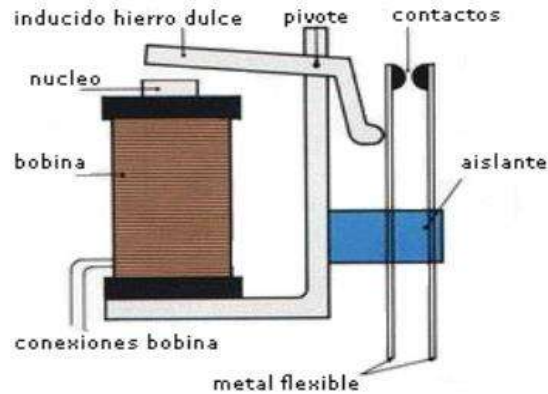


Figura 25. Partes del relé.

## ¿QUE ES UN RELÉ Y PARA QUÉ SIRVE?

Un relé es un interruptor que podemos activar mediante una señal eléctrica [30].

El símbolo del relé muestra la bobina y en este caso, un accionador que conmuta entre dos contactos, pero también existen relés de múltiples contactos. Mediante una señal de control de poca intensidad que excite la bobina podemos conmutar grandes tensiones o intensidades.

¿Por qué usar relés en lugar de transistores?

- Hay límites en la corriente que un transistor puede aceptar, pero un relé se puede diseñar para que aguante cualquier carga, porque basta con que los extremos metálicos de los contactos lo soporten.
- Aísla completamente el circuito de control del de potencia, lo que tiene su importancia especialmente en líneas de media y alta tensión.

Normalmente usaremos un **relé** cuando se requiera conmutar grandes picos de tensión o intensidad como por ejemplo arrancando motores de corriente alterna de una cierta potencia. En cambio **el transistor** es preferible como conmutador, para pequeñas cargas y cuando la velocidad de conmutación sea una cuestión importante.

- Un transistor conmuta varios millones de veces más rápido que un relé.

En la práctica, es más sencillo utilizar un relé para encender una luz fluorescente o la calefacción, que buscar un transistor de características adecuadas.

Aunque hay relés que necesitan muy poca potencia para excitar la bobina, por regla general la salida digital del sistema de control se puede quedar en

corto y proteger sus salidas tendremos que usar un transistor que nos resuelva el problema.

### **El transistor:**

- Cuando ponemos un valor lógico alto en el pin de control, El transistor pasa a saturación y la corriente entre emisor y colector excita la bobina del relé, haciendo que el contacto cambie de posición (y haciendo además un clic).
- Si ponemos lógico bajo en el pin de control el transistor entra en corte e impide el flujo de corriente por lo que la bobina cae y el contacto de salida vuelve a su posición de reposo.

### **CONTACTOR TRIFÁSICO**

El principio de funcionamiento de un contactor como se muestra en la Figura 26 es muy similar al de un relevador o relé eléctrico. Los contactores y los relés eléctricos son interruptores electromagnéticos. Las partes fundamentales de un contactor son:

- Electroimán (bobina más núcleo de hierro).
- Armadura.
- Muelle o resorte.
- Contactos.

### **FUNCIONAMIENTO**

Cuando la bobina se energiza se crea un campo magnético que atrae a la armadura, lo que causa que los contactos se cierren; como resultado se cierra el circuito del motor y este enciende.

Al igual que un relé electromagnético, la bobina del contactor es energizada por un voltaje relativamente menor que el voltaje que es controlado a través de sus contactos. Por lo que los contactores son interruptores eléctricos que con un voltaje realmente bajo pueden controlar equipos que requieran alto voltaje para su funcionamiento (como el motor trifásico).

Cuando la corriente deja de circular a través de la bobina el campo magnético colapsa, por lo que la armadura ya no es atraída y la fuerza producida por el muelle la hacen retornar a la posición inicial. Como resultado los contactos se abren y el motor se apaga.

### **DIFERENCIAS**

La mayor diferencia entre relés y contactores es que los contactores tienen contactos que soportan mayor carga.



Figura 26. Contactor trifásico 220v.

#### 4.2.1.5. Alarma para cal

La alarma para cal nos indicará cuando el agua haya llegado a su punto óptimo de ebullición; como el agregado de cal lo sigue realizando el personal, se colocó un foco el cual a distancia se podrá ver, lo que nos indica que es momento de poner la cal al agua caliente. El control seguirá haciendo lo programado en el sistema de control hasta terminar el proceso de nixtamalización.

### 4.3. Módulo de comunicación del bluetooth

#### ¿QUE ES BLUETOOTH?

Es una especificación tecnológica para redes inalámbricas que permite la transmisión de voz y datos entre distintos dispositivos mediante una radiofrecuencia segura (2.4GHz). El termino Bluetooth (“Diente azul” en inglés, aunque el nombre proviene del rey danés y noruego Harald Blatand, traducido como Harold Bluetooth) es la denominación comercial y popular del estándar de comunicación inalámbrica IEEE.802.15.1. La primera empresa en investigar esta tecnología fue Ericsson, encargada de liderar un grupo que, con el tiempo, sumo a IBM, Nokia, Microsoft, Motorola y otras compañías que apoyaron el estándar. Existen tres clases de bluetooth:

- Clase 1: con un alcance aproximado de 100 metros.
- Clase 2: 10 metros.
- Clase 3: 1 metro.

Los especialistas consideran que, en los próximos años, todos los equipos tecnológicos tendrán la capacidad de comunicarse entre sí gracias al estándar [31]

## MODULO BLUETOOTH HC-05

El módulo Bluetooth HC-05 como en la Figura 27 nos permite conectar nuestros proyectos con Arduino a un Smartphone, celular o PC de forma inalámbrica (Bluetooth), con la facilidad de operación de un puerto serial. La transmisión se realiza totalmente en forma transparente al programador, por lo que se conecta en forma directa a los pines seriales de nuestro microcontrolador preferido (respetando los niveles de voltaje, ya que el módulo se alimenta con 3.3V). La placa también incluye un regulador de 3.3V, que permite alimentar el módulo con un voltaje entre 3.6 a 6V. Este módulo es el complemento ideal para nuestro proyecto de control con Arduino.



Figura 27. Modulo bluetooth HC-05 [7].

## 4.4. Aplicación en App inventor

Es un programa que se puede descargar y al que puede acceder directamente desde su teléfono o desde algún otro aparato móvil como por ejemplo: una Tablet o un reproductor MP3.

### ¿QUE NECESITO PARA DESCARGAR Y USAR UNA APLICACIÓN?

Usted necesita un Smartphone o algún otro aparato móvil con acceso a internet. No todas las aplicaciones funcionan en todos los aparatos móviles. Cuando usted compra uno de estos aparatos debe usar el sistema operativo y el tipo de aplicaciones que corresponde a ese aparato. Los sistemas operativos móviles Android, Apple, Microsoft y BlackBerry tienen tiendas de aplicaciones que operan en línea en las cuales usted puede buscar, descargar e instalar las aplicaciones [32].

### 4.4.1. App Inventor

#### Flujo de ejecución de una aplicación en App Inventor

El flujo de cualquier programa en Android se basa en eventos Figura 28, es decir, tiene que ocurrir una acción para que la reacción correspondiente se realice, es decir, el programa

“espera” a. Esto ocurre tanto en las aplicaciones generadas con la programación “tradicional” como las generadas con App Inventor [33].

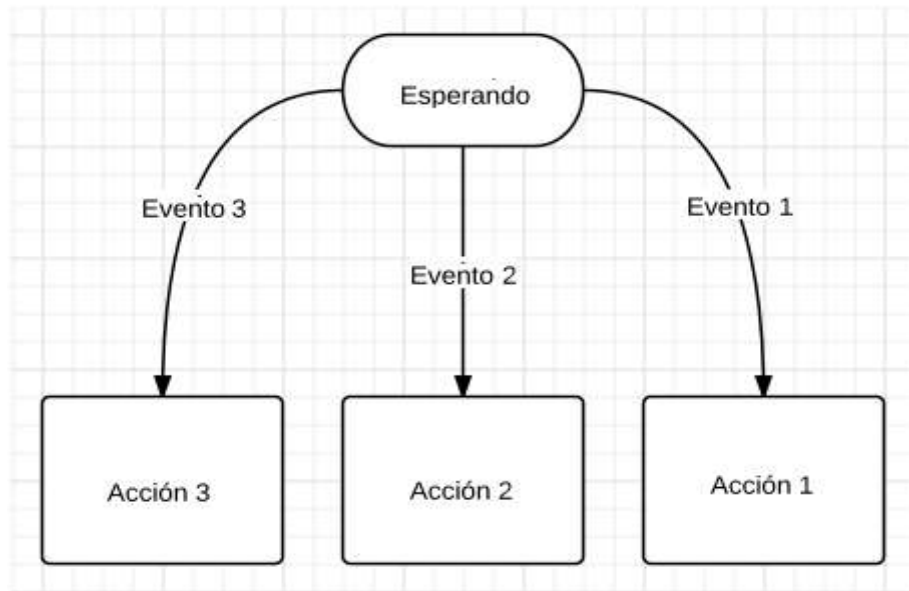


Figura 28. Diagrama de flujo en App inventor.

En el caso de la programación en otros lenguajes, Arduino es una plataforma Figura 29, la programación se basa en un ciclo en el que de manera lineal se ejecuta el código. Dentro de este ciclo, tendremos que comprobar si las acciones han ocurrido para entonces ejecutar las acciones correspondientes.

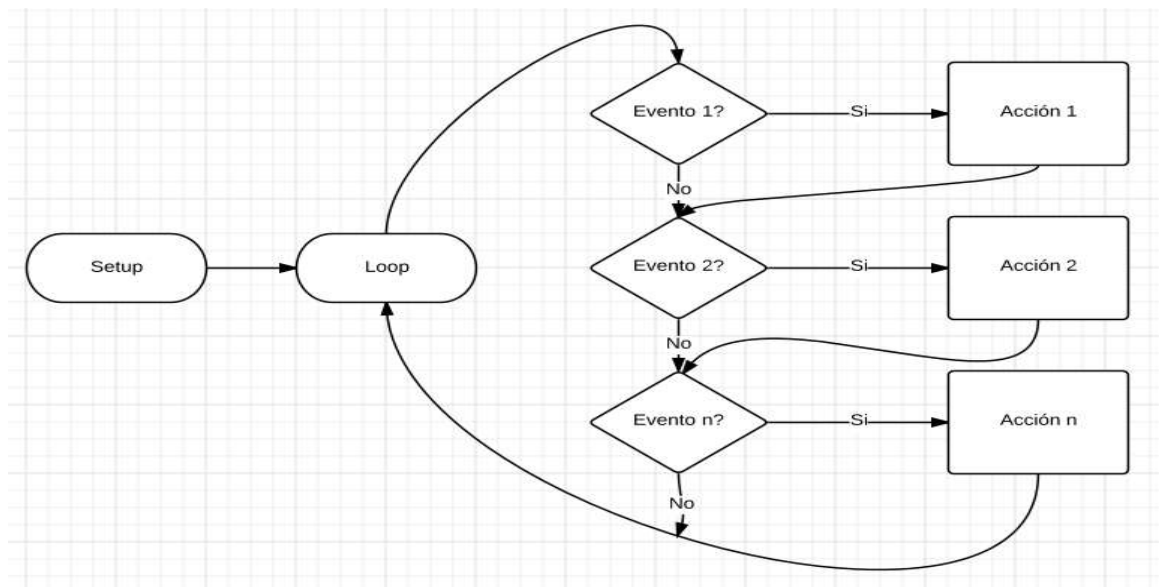


Figura 29. Diagrama de flujo de Arduino.

La plataforma para desarrollo de aplicaciones App inventor, es muy amigable con la gente que quiere aprender y desarrollar una aplicación, tiene muchas herramientas para realizar el código.

Por ello, la decisión de realizar la aplicación de la “AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN”, ya que no requiere de mucha sofisticación para realizar el prototipo; como también la fácil utilización del usuario al contener solo lo necesario.

Esta es la pantalla principal de la aplicación. Como se puede ver en la Figura 30 en la columna izquierda, se encuentra todo lo que se puede utilizar para crear una aplicación como la interfaz de usuario, medios, conectividad, etc. Y en la columna derecha podemos modificar las características de lo que se haya seleccionado para la aplicación.

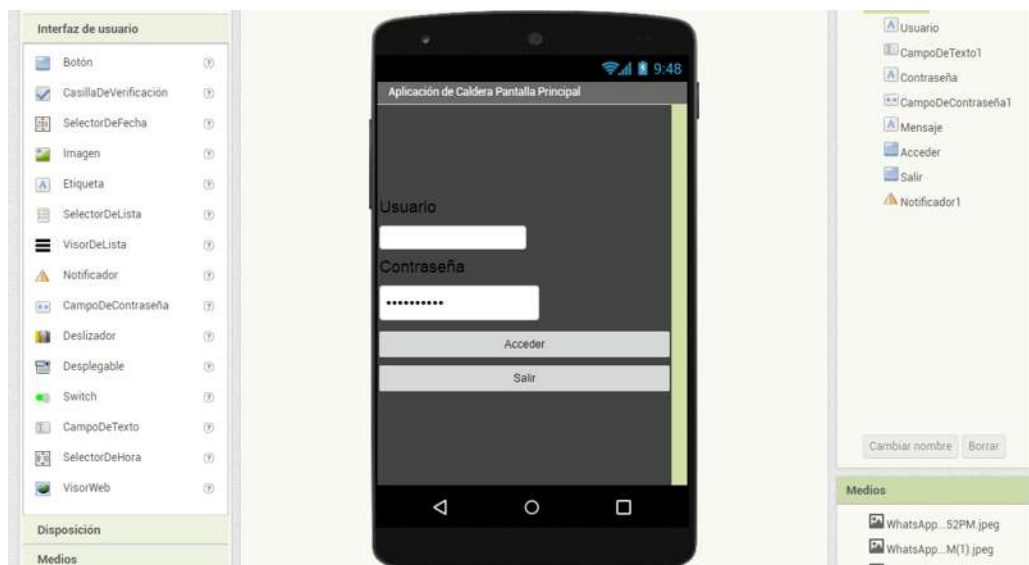


Figura 30. Vista de componentes de la plataforma en App inventor [7].

En la Figura 31 se muestra el código realizado a bloques para la pantalla principal de la aplicación. El cual cómo podemos ver, los bloques están hechos por ciclos, sentencias y condiciones para su funcionamiento, lo cual es muy práctico para los que quieren realizar su primer aplicación en **App inventor**.

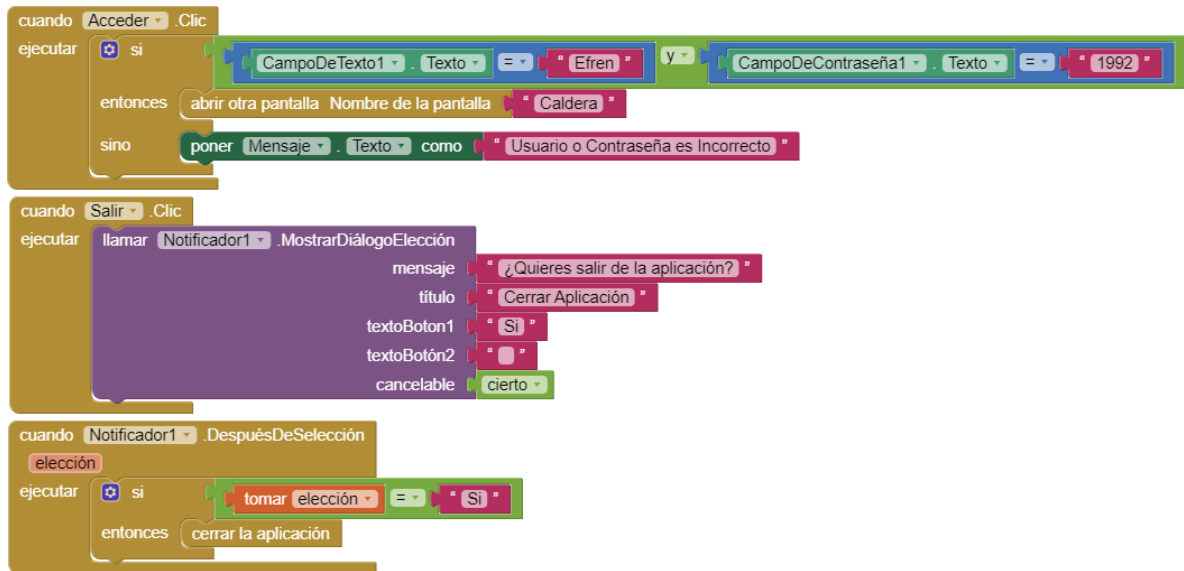


Figura 31. Código desarrollado para el ingreso a la aplicación [7].

Una vez ingresado el usuario y contraseña, podremos ingresar al menú principal de la aplicación, para la AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN, vista desde la página web de App inventor.

Como se puede observar en la Figura 32 en la columna izquierda, se encuentran todos los componentes que se pueden utilizar para crear la aplicación, en la columna derecha, se encuentran todos los componentes seleccionados, para modificar, cambiar tipos de letra, tamaños y renombrar, como también colocar imágenes de fondo a la aplicación.

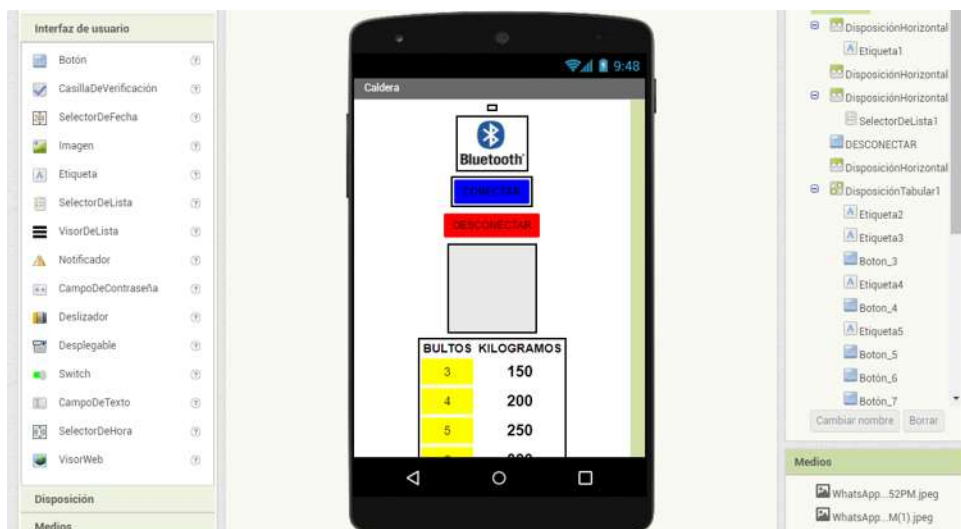


Figura 32. Menú de pantalla principal de la AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN.

En la Figura 33 se encuentra desarrollado el código a bloques, para la aplicación de AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN. Para cada uno de los botones se tiene que programar con sentencias, condiciones, ciclos, pero no es de código abierto. Se encuentra una columna con lógicas, textos, control, procedimientos que pueden ser utilizados para cada uno de los elementos puestos en la aplicación.

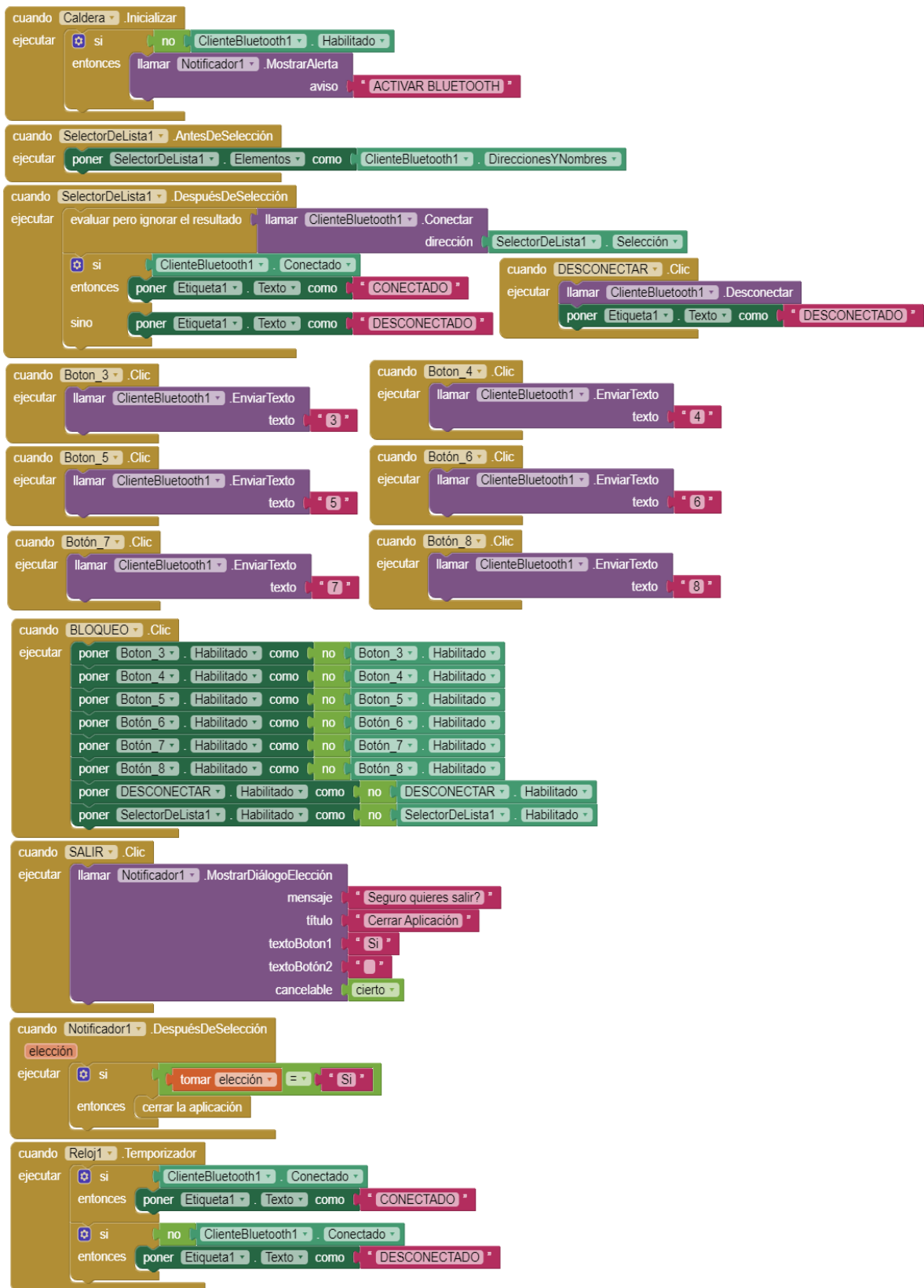


Figura 33. Código del menú principal de la aplicación Android.

## 4.4.2. Desarrollo del Código en la plataforma Arduino

Este código es el que controla los actuadores y recibe datos mediante los sensores, por lo cual el código desarrollado es el que está controlando nuestro proceso de nixtamalización, una vez se haya recibido el dato enviado por la aplicación de **App Inventor**.

### CÓDIGO:

```
#include <DallasTemperature.h>
#include <OneWire.h>

const int pinDatosDQ = 7;
OneWire oneWireObjeto(pinDatosDQ);
DallasTemperature sensorDS18B20(&oneWireObjeto);

const int Trigger = 5; //Pin digital 2 para el Trigger del sensor
const int Echo = 6; //Pin digital 3 para el Echo del sensor
const char dato = LOW;
long t; //timepo que demora en llegar el eco
long d; //distancia en centimetros
long cm;
const int agua = 13; //pin de agua
const int gas = 12; //pin de gas
const int chispa = 11; //pin de chispa modificar
const int motor = 10; //pin de motor
const int comu = 9; //pin de comunicación serial
const int temp = 8; //pin de temperatura Led

void setup()
{
  Serial.begin(9600);//iniciaizamos la comunicación
  sensorDS18B20.begin(); //se inicia el bus 1-Wire
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT); //pin como entrada
  digitalWrite(Trigger, LOW);//Inicializamos el pin con 0
  pinMode(agua,OUTPUT); //pin salida para valvula de agua
  pinMode(gas,OUTPUT); //pin salida para valvula de Gas
  digitalWrite(gas,LOW);
  pinMode(chispa,OUTPUT); //pin salida para generador de Chispa
  digitalWrite(chispa,LOW);
  pinMode(motor,OUTPUT); //pin salida para motor
  digitalWrite(motor,LOW);
  pinMode(comu,OUTPUT); //pin de comunicación
  pinMode(temp,OUTPUT); //pin de led indicador de temperatura
}

char menu ()
{
  if (Serial.available(>0)
  {
    digitalWrite(comu,HIGH); //pin de comunicación
    //programa principal
```

```

while(1)
{
  char dato = Serial.read(); //leer el puerto serie y guardar datos en la variable dato

  switch(dato) // caso de selección
  {
    case '3': // cuando sea 3 en nuestro caso es la cantidad de bultos a ingresar

      while(dato == '3')
      {
        digitalWrite(agua,HIGH);
        digitalWrite(Tripwire, HIGH);
        delayMicroseconds(10); //Enviamos un pulso de 10us
        digitalWrite(Tripwire, LOW);
        t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
        d = t/59; //escalamos el tiempo a una distancia en cm
        Serial.print("Distancia: ");
        Serial.print(d); //Enviamos serialmente el valor de la distancia
        Serial.print("cm");
        Serial.println();
        delay(1000); //Hacemos una pausa de 100ms

        while (d < 15)
        {
          digitalWrite(agua,LOW); //cerramos valvula de agua
          digitalWrite(gas,HIGH); //valvula de gas abierta
          digitalWrite(chispa,HIGH); //generador de chispa prendido
          delay(2000);
          digitalWrite(chispa,LOW); //generador de chispa prendido
          delay(1000);
          digitalWrite(chispa,HIGH); //generador de chispa prendido
          delay(1000);
          digitalWrite(chispa,LOW); //generador de chispa prendido
          delay(1000);
          digitalWrite(chispa,HIGH); //generador de chispa prendido
          delay(1000);
          digitalWrite(chispa,LOW); //generador de chispa prendido

          while(true)
          {
            sensorDS18B20.requestTemperatures();
            Serial.print("Temperatura sensor: ");
            Serial.print(sensorDS18B20.getTempCByIndex(0));
            Serial.println(" °C");

            digitalWrite(chispa,LOW);

            if (sensorDS18B20.getTempCByIndex(0) > 15 && sensorDS18B20.getTempCByIndex(0) < 50)
            {
              digitalWrite(temp,HIGH);
              delay(100);
              digitalWrite(temp,LOW);
            }
          }
        }
      }
    }
  }
}

```

```

delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
}
if (sensorDS18B20.getTempCByIndex(0) > 50 && sensorDS18B20.getTempCByIndex(0) < 80)
{
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
}

while (sensorDS18B20.getTempCByIndex(0) > 91 && sensorDS18B20.getTempCByIndex(0) < 95)
{
digitalWrite(temp,HIGH); //temperatura encendido
digitalWrite(gas,LOW); //valvula de gas cerrada
digitalWrite(motor,HIGH); // Motor encendido
sensorDS18B20.requestTemperatures();
Serial.print("Temperatura sensor: ");
Serial.print(sensorDS18B20.getTempCByIndex(0));
Serial.println(" °C");
delay(1000);

digitalWrite(Trip, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us
digitalWrite(Trip, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
cm = t/59; //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(cm); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(1000); //Hacemos una pausa de 100ms

while(cm < 13)
{
digitalWrite(chispa,LOW);
digitalWrite(motor,LOW);
digitalWrite(temp,LOW);
return 0;
}
}
}
} //termina d menor a 15
} //termina while(dato)

```

```

break; // termina el caso (3)

case '4': // cuando sea 3 en nuestro caso es la cantidad de bultos a ingresar

while(dato == '4')
{
digitalWrite(agua,HIGH);
digitalWrite(Trip, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us
digitalWrite(Trip, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
d = t/59; //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(d); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(1000); //Hacemos una pausa de 100ms

while (d < 14)
{
digitalWrite(agua,LOW); //cerramos valvula de agua
digitalWrite(gas,HIGH); //valvula de gas abierta
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(2000);
digitalWrite(chispa,LOW); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,LOW); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,LOW); //generador de chispa prendido

while(true)
{
sensorDS18B20.requestTemperatures();
Serial.print("Temperatura sensor: ");
Serial.print(sensorDS18B20.getTempCByIndex(0));
Serial.println(" °C");

digitalWrite(chispa,LOW);

if (sensorDS18B20.getTempCByIndex(0) > 15 && sensorDS18B20.getTempCByIndex(0) < 50)
{
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
}
}
}

```

```

}
if (sensorDS18B20.getTempCByIndex(0) > 50 && sensorDS18B20.getTempCByIndex(0) < 80)
{
  digitalWrite(temp,HIGH);
  delay(100);
  digitalWrite(temp,LOW);
  delay(100);
  digitalWrite(temp,HIGH);
  delay(100);
  digitalWrite(temp,LOW);
  delay(100);
  digitalWrite(temp,HIGH);
  delay(100);
  digitalWrite(temp,LOW);
}

while (sensorDS18B20.getTempCByIndex(0) > 91 && sensorDS18B20.getTempCByIndex(0) < 95)
{
  digitalWrite(temp,HIGH); //temperatura encendido
  digitalWrite(gas,LOW); //valvula de gas cerrada
  digitalWrite(motor,HIGH); // Motor encendido
  Serial.print("Temperatura sensor: ");
  Serial.print(sensorDS18B20.getTempCByIndex(0));
  Serial.println(" °C");
  delay(1000);

  digitalWrite(Trip, HIGH);
  delayMicroseconds(10); //Enviamos un pulso de 10us
  digitalWrite(Trip, LOW);
  t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
  cm = t/59; //escalamos el tiempo a una distancia en cm
  Serial.print("Distancia: ");
  Serial.print(cm); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(1000); //Hacemos una pausa de 100ms

  while(cm < 12)
  {
    digitalWrite(chispa,LOW);
    digitalWrite(motor,LOW);
    digitalWrite(temp,LOW);
    return 0;
  }
}
}
}
} //termina d menor a 18

} //termina while(dato)
break; // termina el caso (4)

case '5': // cuando sea 3 en nuestro caso es la cantidad de bultos a ingresar

```

```

while(dato == '5')
{
digitalWrite(agua,HIGH);
digitalWrite(Trigger, HIGH);
delayMicroseconds(10);    //Enviamos un pulso de 10us
digitalWrite(Trigger, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
d = t/59;    //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(d);    //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(1000);    //Hacemos una pausa de 100ms

while (d < 13)
{
digitalWrite(agua,LOW); //cerramos valvula de agua
digitalWrite(gas,HIGH); //valvula de gas abierta
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(2000);
digitalWrite(chispa,LOW); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,LOW); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,LOW); //generador de chispa prendido

while(true)
{
sensorDS18B20.requestTemperatures();
Serial.print("Temperatura sensor: ");
Serial.print(sensorDS18B20.getTempCByIndex(0));
Serial.println(" °C");

digitalWrite(chispa,LOW);

if (sensorDS18B20.getTempCByIndex(0) > 15 && sensorDS18B20.getTempCByIndex(0) < 50)
{
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
}
if (sensorDS18B20.getTempCByIndex(0) > 50 && sensorDS18B20.getTempCByIndex(0) < 80)
{
digitalWrite(temp,HIGH);

```

```

delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
}

while (sensorDS18B20.getTempCByIndex(0) > 91 && sensorDS18B20.getTempCByIndex(0) < 95)
{
digitalWrite(temp,HIGH); //temperatura encendido
digitalWrite(gas,LOW); //valvula de gas cerrada
digitalWrite(motor,HIGH); // Motor encendido
Serial.print("Temperatura sensor: ");
Serial.print(sensorDS18B20.getTempCByIndex(0));
Serial.println(" °C");
delay(1000);

digitalWrite(Trip, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us
digitalWrite(Trip, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
cm = t/59; //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(cm); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(1000); //Hacemos una pausa de 100ms

while(cm < 11)
{
digitalWrite(chispa,LOW);
digitalWrite(motor,LOW);
digitalWrite(temp,LOW);
return 0;
}
}
} //termina d menor a 18
} //termina while(dato)
break; // termina el caso (5)

case '6': // cuando sea 3 en nuestro caso es la cantidad de bultos a ingresar

while(dato == '6')
{
digitalWrite(agua,HIGH);
digitalWrite(Trip, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us

```

```

digitalWrite(Trip, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
d = t/59; //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(d); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(1000); //Hacemos una pausa de 100ms

while (d < 12)
{
digitalWrite(agua,LOW); //cerramos valvula de agua
digitalWrite(gas,HIGH); //valvula de gas abierta
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(2000);
digitalWrite(chispa,LOW); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,LOW); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,LOW); //generador de chispa prendido

while(true)
{
sensorDS18B20.requestTemperatures();
Serial.print("Temperatura sensor: ");
Serial.print(sensorDS18B20.getTempCByIndex(0));
Serial.println(" °C");

digitalWrite(chispa,LOW);

if (sensorDS18B20.getTempCByIndex(0) > 15 && sensorDS18B20.getTempCByIndex(0) < 50)
{
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
}
if (sensorDS18B20.getTempCByIndex(0) > 50 && sensorDS18B20.getTempCByIndex(0) < 80)
{
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
}
}
}

```

```

digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
}

while (sensorDS18B20.getTempCByIndex(0) > 91 && sensorDS18B20.getTempCByIndex(0) < 95)
{
digitalWrite(temp,HIGH); //temperatura encendido
digitalWrite(gas,LOW); //valvula de gas cerrada
digitalWrite(motor,HIGH); // Motor encendido
Serial.print("Temperatura sensor: ");
Serial.print(sensorDS18B20.getTempCByIndex(0));
Serial.println(" °C");
delay(1000);

digitalWrite(Trip, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us
digitalWrite(Trip, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
cm = t/59; //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(cm); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(1000); //Hacemos una pausa de 100ms

while(cm < 10)
{
digitalWrite(chispa,LOW);
digitalWrite(motor,LOW);
digitalWrite(temp,LOW);
return 0;
}
}
} //termina d menor a 18
} //termina while(dato)
break; // termina el caso (6)

case '7': // cuando sea 3 en nuestro caso es la cantidad de bultos a ingresar

while(dato == '7')
{
digitalWrite(agua,HIGH);
digitalWrite(Trip, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us
digitalWrite(Trip, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
d = t/59; //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(d); //Enviamos serialmente el valor de la distancia

```

```

Serial.print("cm");
Serial.println();
delay(1000);    //Hacemos una pausa de 100ms

while (d < 11)
{
  digitalWrite(agua,LOW); //cerramos valvula de agua
  digitalWrite(gas,HIGH); //valvula de gas abierta
  digitalWrite(chispa,HIGH); //generador de chispa prendido
  delay(2000);
  digitalWrite(chispa,LOW); //generador de chispa prendido
  delay(1000);
  digitalWrite(chispa,HIGH); //generador de chispa prendido
  delay(1000);
  digitalWrite(chispa,LOW); //generador de chispa prendido
  delay(1000);
  digitalWrite(chispa,HIGH); //generador de chispa prendido
  delay(1000);
  digitalWrite(chispa,LOW); //generador de chispa prendido

  while(true)
  {
    sensorDS18B20.requestTemperatures();
    Serial.print("Temperatura sensor: ");
    Serial.print(sensorDS18B20.getTempCByIndex(0));
    Serial.println(" °C");

    digitalWrite(chispa,LOW);

    if (sensorDS18B20.getTempCByIndex(0) > 15 && sensorDS18B20.getTempCByIndex(0) < 50)
    {
      digitalWrite(temp,HIGH);
      delay(100);
      digitalWrite(temp,LOW);
      delay(100);
      digitalWrite(temp,HIGH);
      delay(100);
      digitalWrite(temp,LOW);
    }
    if (sensorDS18B20.getTempCByIndex(0) > 50 && sensorDS18B20.getTempCByIndex(0) < 80)
    {
      digitalWrite(temp,HIGH);
      delay(100);
      digitalWrite(temp,LOW);
      delay(100);
      digitalWrite(temp,HIGH);
      delay(100);
      digitalWrite(temp,LOW);
      delay(100);
      digitalWrite(temp,HIGH);
      delay(100);
      digitalWrite(temp,LOW);
    }
  }
}

```

```

}

while (sensorDS18B20.getTempCByIndex(0) > 91 && sensorDS18B20.getTempCByIndex(0) < 95)
{
  digitalWrite(temp,HIGH); //temperatura encendido
  digitalWrite(gas,LOW); //valvula de gas cerrada
  digitalWrite(motor,HIGH); // Motor encendido
  Serial.print("Temperatura sensor: ");
  Serial.print(sensorDS18B20.getTempCByIndex(0));
  Serial.println(" °C");
  delay(1000);

  digitalWrite(Trip, HIGH);
  delayMicroseconds(10); //Enviamos un pulso de 10us
  digitalWrite(Trip, LOW);
  t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
  cm = t/59; //escalamos el tiempo a una distancia en cm
  Serial.print("Distancia: ");
  Serial.print(cm); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(1000); //Hacemos una pausa de 100ms

  while(cm < 9)
  {
    digitalWrite(chispa,LOW);
    digitalWrite(motor,LOW);
    digitalWrite(temp,LOW);
    return 0;
  }
}
} //termina d menor a 18
} //termina while(dato)
break; // termina el caso (7)

case '8': // cuando sea 3 en nuestro caso es la cantidad de bultos a ingresar

while(dato == '8')
{
  digitalWrite(agua,HIGH);
  digitalWrite(Trip, HIGH);
  delayMicroseconds(10); //Enviamos un pulso de 10us
  digitalWrite(Trip, LOW);
  t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
  d = t/59; //escalamos el tiempo a una distancia en cm
  Serial.print("Distancia: ");
  Serial.print(d); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(1000); //Hacemos una pausa de 100ms
}

```

```

while (d < 10)
{
digitalWrite(agua,LOW); //cerramos valvula de agua
digitalWrite(gas,HIGH); //valvula de gas abierta
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(2000);
digitalWrite(chispa,LOW); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,LOW); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,HIGH); //generador de chispa prendido
delay(1000);
digitalWrite(chispa,LOW); //generador de chispa prendido

while(true)
{
sensorDS18B20.requestTemperatures();
Serial.print("Temperatura sensor: ");
Serial.print(sensorDS18B20.getTempCByIndex(0));
Serial.println(" °C");

digitalWrite(chispa,LOW);

if (sensorDS18B20.getTempCByIndex(0) > 15 && sensorDS18B20.getTempCByIndex(0) < 50)
{
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
}
if (sensorDS18B20.getTempCByIndex(0) > 50 && sensorDS18B20.getTempCByIndex(0) < 80)
{
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
delay(100);
digitalWrite(temp,HIGH);
delay(100);
digitalWrite(temp,LOW);
}

while (sensorDS18B20.getTempCByIndex(0) > 91 && sensorDS18B20.getTempCByIndex(0) < 95)
{
digitalWrite(temp,HIGH); //temperatura encendido

```

```

digitalWrite(gas,LOW); //valvula de gas cerrada
digitalWrite(motor,HIGH); // Motor encendido
Serial.print("Temperatura sensor: ");
Serial.print(sensorDS18B20.getTempCByIndex(0));
Serial.println(" °C");
delay(1000);

digitalWrite(Triiger, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us
digitalWrite(Triiger, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
cm = t/59; //escalamos el tiempo a una distancia en cm
Serial.print("Distancia: ");
Serial.print(cm); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(1000); //Hacemos una pausa de 100ms

while(cm < 8)
{
digitalWrite(chispa,LOW);
digitalWrite(motor,LOW);
digitalWrite(temp,LOW);
return 0;
}
}
} //termina d menor a 18
} //termina while(dato)
break; // termina el caso (8)

} // termina el swicht(dato)
} //termina while principal
} //termina comunicacion
} // termina void menu

void loop()
{
menu();
}

```

# Capítulo 5 Prototipo del sistema a desarrollar

## 5.1. El Arduino como elemento de control

### 5.1.1. Descripción del control

Inicialmente, teniendo el maíz en el depósito y con la cantidad deseada, seleccionamos la cantidad que se requiere poner a cocer, por medio de la aplicación desarrollada en App inventor.

Una vez que el sistema de control recibe la orden de iniciar desde la aplicación Android, toma la decisión de abrir la válvula de agua, medirá el nivel de agua por medio del sensor ultrasónico, el cual indica el nivel de agua. Al alcanzar el límite seleccionado del sistema de control envía la acción de cerrar la válvula de agua.

Simultáneamente se activa la válvula de gas, permitiendo la salida de gas y activa al generador de chispa, el cual encenderá el gas.

El sensor de temperatura monitorea la temperatura del agua; alcanza la temperatura adecuada el sistema de control desactiva la electroválvula de gas y activa el contactor para que el motor suba el maíz del depósito hacia la caldera con agua caliente y cal disuelta.

Al agregar el maíz al agua el volumen en la caldera aumenta, por lo que usaremos este fenómeno para determinar cuándo se haya vaciado todo el maíz del depósito a la caldera. Por lo cual medimos con el sensor ultrasónico el nivel de la caldera y cuando este nivel no cambie nos indica que se ha terminado de vaciar el depósito de maíz en la caldera. En ese momento el sistema de control desactiva el motor y se finaliza el control.

## 5.1.2. Diagrama de flujo del software del microcontrolador.

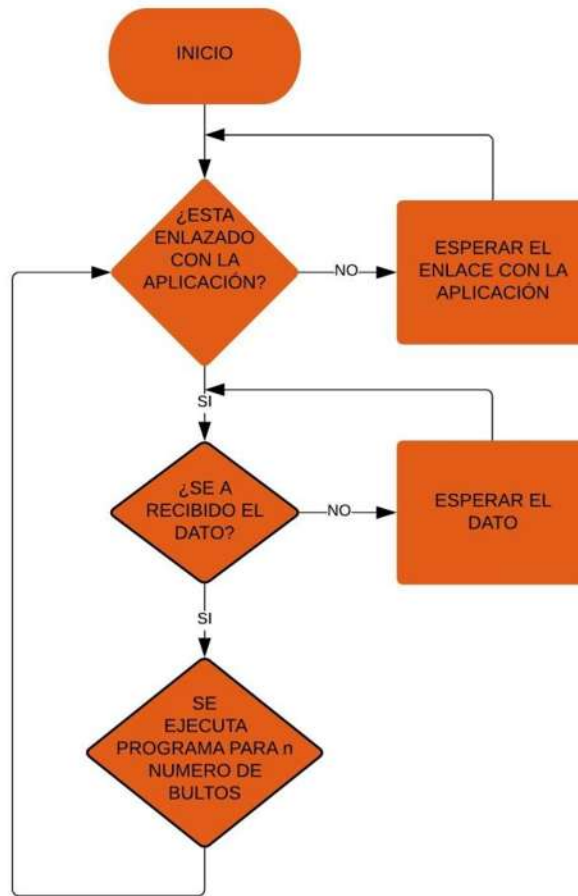


Figura 34. Diagrama de flujo del código de Arduino [7]

## 5.2. Aplicación para control de Arduino

En la Figura 35 se muestra la pantalla principal en donde se debe introducir un nombre de usuario y contraseña para poder ingresar.

Al ingresar aparece un menú principal Figura 36 de la aplicación.

La aplicación está desarrollada para dispositivos Android y está aplicada en un celular real.



Figura 35. Pantalla principal de la aplicación en dispositivo Android [7].



Figura 36. Pantalla del menú para la AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN [7].

### 5.2.1. Descripción de la App

Para utilizar la aplicación primeramente accederemos a la aplicación de la caldera, nos aparecerá la pantalla principal, en la cual se ingresa el nombre de usuario y contraseña ya pre-establecido, si no se cuenta con el usuario ni contraseña no se podrá ingresar a la aplicación

En la segunda pantalla una vez puesto usuario y contraseña correcta nos aparecerá un menú, el cual está diseñado con lo necesario para el control de la **AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN**.

Primer paso: conectarnos con el dispositivo bluetooth, para ello pulsamos el recuadro que dice conectar, nos arroja una lista con todos los dispositivos bluetooth guardados en el dispositivo, buscamos el nombre del bluetooth **HC-05** y lo seleccionamos.

Segundo paso: una vez conectados ya podemos elegir el número de bultos que queremos nixtamalizar eligiendo la cantidad marcada en los botones amarillos.

Tercer paso: una vez empezado el proceso de nixtamalización por seguridad y no se presione nuevamente otro número, presionamos el botón de bloqueo de botones que se encuentra de color rojo.

Cuarto paso: salir de la aplicación si ya no se requiere nixtamalizar más maíz.

## 5.2.2. Diagrama de flujo del Software de la App

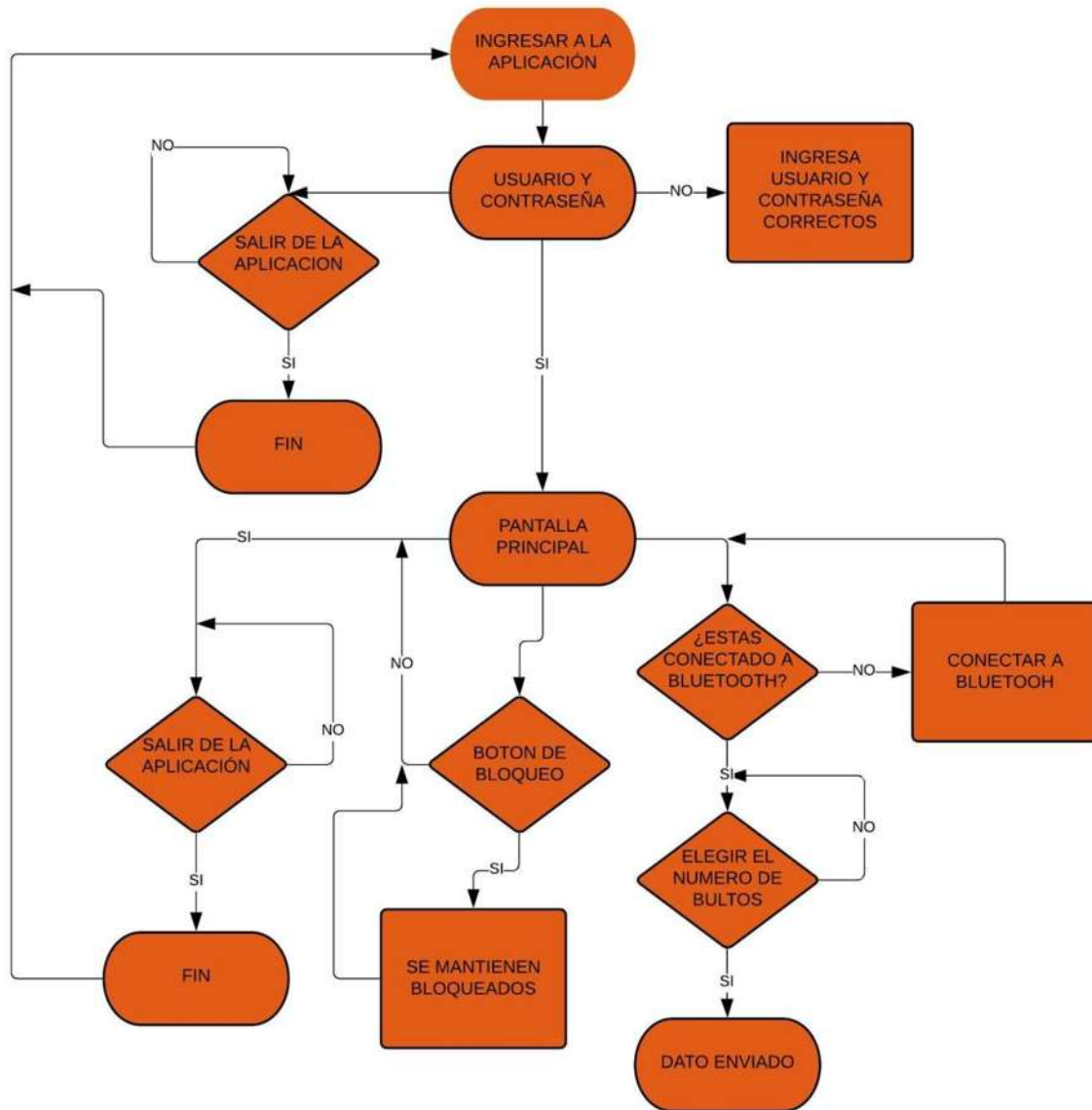


Figura 37. Diagrama de flujo del código realizado para la aplicación Android [7].

# Capítulo 6 Prototipo desarrollado, pruebas y resultados

## 6.1. Esquemático

En la Figura 38 se muestra la forma de conexión del sensor DS18B20 hacia el **Arduino Uno**. Como también si se requiere conectar más de un sensor al mismo pin de entrada.

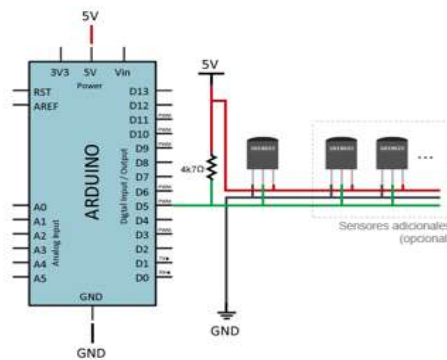


Figura 38. Conexión del sensor de temperatura.

En la Figura 39 se muestra la conexión de un sensor ultrasónico HC-SR04 para recibir los datos e interpretarlos como distancia, para nuestro caso se conectaron en el trigger (pin 2) y echo (pin 3), Vcc a 5v y GND.

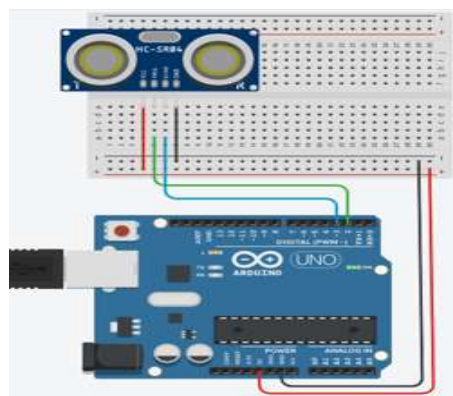


Figura 39. Conexión del sensor ultrasónico.

En la Figura 40 se muestra la conexión del relevador para activar el generador de chispa y motor trifásico que está encendiendo por medio de un contactor que a su vez se activa el relevador y protegerá al **Arduino Uno**. Para este caso necesitaremos de dos conexiones

como las del relevador, una para controlar el contactor y otra para la generación de la chispa.

La salida para el generador de chispa se encuentra en el pin 11.

La salida para el motor se encuentra en el pin 10.

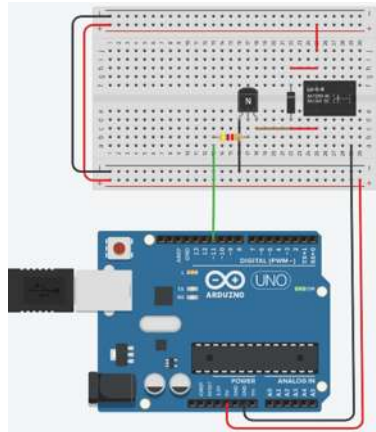


Figura 40. Conexión de relevador.

En la Figura 41 se muestra con leds las salidas digitales específicas para el uso de este prototipo la cual está conformada por todos los sensores y actuadores que se usaran.

- El led azul controla la electroválvula de agua (pin 13).
- El led amarillo controla la electroválvula de gas (pin 12).
- El led verde controla el generador de chispa (pin 11).
- El led blanco controla el motor (pin 10).
- El led naranja nos indica si está disponible el serial (pin 9).
- El led rojo nos indica la temperatura al parpadear y cuando esté la temperatura requerida se queda encendido (pin 8).

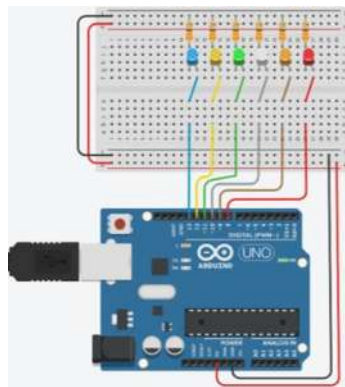
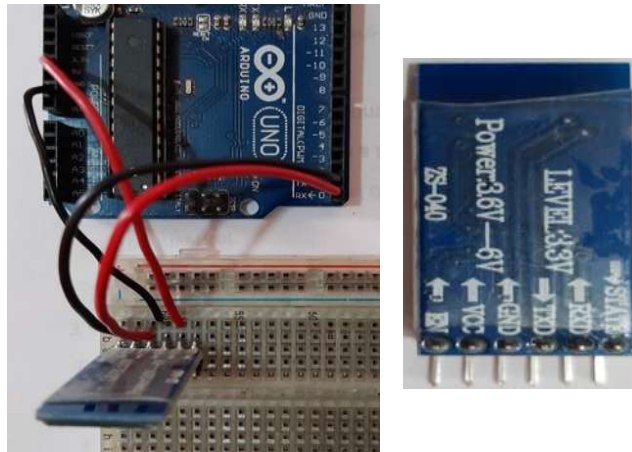


Figura 41. Salidas digitales para el control de sensores y actuadores.

En la Figura 42 se muestra la conexión del módulo de bluetooth al **Arduino Uno** el cual su salida TX y entrada RX están preestablecidas y en el módulo de la misma manera solo quedaría conectar a dichos pines.



*Figura 42. Conexión del módulo bluetooth*

En la Figura 43 se muestra todo el prototipo conectado a la Placa Arduino Uno a nivel representativo.



Figura 43. AUTOMATIZACIÓN DEL PROCESO DE NIXTAMALIZACIÓN [7].

## 6.2. Pruebas y resultados

Los resultados fueron los esperados, aunque existen limitantes que no pueden pasar omitidas ya que por ello el prototipo puede ser más seguro y preciso para el desarrollo de la automatización, después de varios intentos y varias medicaciones se ajustó el código a las necesidades del prototipo, las pruebas fueron hechas para cada uno de los bultos a nixtamalizar y comprobar su correcto funcionamiento.

La aplicación fue puesta a prueba por un usuario el cual ingresó al menú y eligió una cantidad de bultos al azar y los resultados fueron los esperados, su correcto funcionamiento de la aplicación y del dato enviado hacia el sistema de control, el cual inicia el proceso seleccionado.

# Capítulo 7 Conclusiones, recomendaciones y trabajos futuros.

## 7.1. Conclusiones

Este prototipo es la primera versión de una automatización, que puede ser mejorada en base a lo ya establecido, haciendo que el proceso de nixtamalización, que se hace por medio de la supervisión del personal, se pueda automatizar con el uso de la nueva tecnología.

Este prototipo, innovando el proceso prehispánico, donde podamos alcanzar un proceso de nixtamalización más exacto y una calidad en su producción de materia prima mucho mayor.

En la realización de esta tesis me encontré con muchas dificultades, la cual para lo que presenta a simple perspectiva o parecido se puede decir fácil pero con forme me iba adentrando al tema fui analizando que las cosas no solo eran así de sencillas, existen muchas limitantes y muchas cosas que deben ser tomadas en cuenta que no se pueden cambiar, a menos que se encontrara alguna mejor forma de hacer la nixtamalización.

La principal dificultad que me enfrente fue en la realización del código en **Arduino Uno** el cual era totalmente nuevo para mí, tuve que estudiar cómo funcionaba su desarrollo de código como lo interpretaba y como escribir en el lenguaje que se utiliza para **Arduino Uno**, en todos los sentidos. De igual forma la realización de una aplicación para celular me representó un reto por entrar en un mundo totalmente desconocido y nuevo, siempre es muy gratificante crear tu propia Aplicación celular y que a la cual puedas todavía hacerla crecer y sea mucho mejor. El tipo de programación que se usa en la plataforma **App inventor** es algo que nunca había programado en ese tipo de forma.

Fue un reto más que un trabajo para mí, y un logro al concluir esta tesis.

## 7.2. Recomendaciones

Para poder llegar a automatizar completamente el proceso de la nixtamalización es recomendable que se lleve a cabo un reacondicionamiento de los equipos (tina de reposo, caldera, depósito de maíz), hacer que el maíz se encuentre siempre disponible cuando se quiera empezar la nixtamalización sin necesidad de que el trabajador llene un depósito para poder empezar con la nixtamalización. Una vez terminando todo el proceso de nixtamalización, automáticamente se vacíe en la tina de reposo.

Utilizar sensores que no sean tan sensibles a perturbaciones debido a vapores y calentamiento del mismo proceso de nixtamalización. Actuadores de uso industrial con una mayor durabilidad y con resistencia a altas temperaturas.

Módulo de bluetooth con un mayor rango de conectividad.

### 7.3. Trabajos futuros

- Mejorar la aplicación en cuanto a seguridad de conectividad.
- Mejorar el menú principal de la aplicación (cambiar la forma de adquirir los datos).
- Mejorar la conectividad mediante bluetooth.
- Agregar más sensores para controlar con seguridad el proceso de nixtamalización.
- Poder tener una pantalla que nos esté mostrando el proceso de la nixtamalización.
- Poner un medidor de pH y adquirir los datos para posterior usarlos.
- Poder regular la intensidad de la salida de gas por medio de la Aplicación celular.
- Trabajar en las cuestiones de seguridad.

# Bibliografía

- [1] wikipedia, «wikipediaNixtamalización,» [En línea]. Available: <https://bit.ly/2TItRVp>.
- [2] O. y. G. L. F. B. P. L. A. Paredes López, «la nixtamalización y el valor nutritivo del maíz,» *ciencias*, nº <https://bit.ly/2K0TaJl>, pp. 60-70, 92-93.
- [3] SAGARPA, «PLANEACION AGRICOLA NACIONAL,» 2017-2030. [En línea]. Available: <https://bit.ly/3916BCs>.
- [4] «asociación de consumidores orgánicos,» 24 06 2017. [En línea]. Available: <https://consumidoresorganicos.org/2017/06/24/si-hay-nixtamalizacion-hay-mejor-nutricion/>. [Último acceso: 07 Marzo 2020].
- [5] «+DEMEX,» 06 09 2019. [En línea]. Available: <https://masdemx.com/2016/09/6-maravillas-nutricionales-de-la-nixtamalizacion-mexicana-sin-ella-no-son-tortillas/>. [Último acceso: 07 Marzo 2020].
- [6] «del maiz,» 01 2017. [En línea]. Available: <http://delmaiz.info/wp-content/uploads/2017/01/estructura-grano-ma%C3%ADz.jpg>. [Último acceso: 10 Marzo 2020].
- [7] E. Zúñiga Chávez, 02 Marzo 2020. [En línea].
- [8] «TP-Laboratorio Químico,» [En línea]. Available: <https://www.tplaboratorioquimico.com/laboratorio-quimico/materiales-e-instrumentos-de-un-laboratorio-quimico/phmetro.html>. [Último acceso: 09 Marzo 2020].
- [9] J. P. P. y. A. Gardey., «Definición.DE,» 2010. [En línea]. Available: <https://definicion.de/sensor/>. [Último acceso: 02 Marzo 2020].
- [10] E. V. C. y. S. C. S.A., «Actuadores.doc,» [En línea]. Available: <http://www.aie.cl/files/file/comites/ca/abc/actuadores.pdf>. [Último acceso: 02 Marzo 2020].
- [11] «descubrearduino,» [En línea]. Available: <https://descubrearduino.com/arduino-uno/>. [Último acceso: 02 Marzo 2020].
- [12] «Facultad de ingeniería U.N.N.E,» [En línea]. Available: <http://ing.unne.edu.ar/pub/informatica/U2.pdf>. [Último acceso: 02 Marzo 2020].

- [13 «Introducción a la Operación de Computadoras Personales,» [En línea]. Available:  
] <http://cs.uns.edu.ar/materias/iocp/downloads/Clases%20Teoricas/Clase-01-Conceptos-basicos-Hardware.pdf>. [Último acceso: 02 Marzo 2020].
- [14 J. S. López., 11 2013. [En línea]. Available:  
] <https://proyectocirculos.files.wordpress.com/2013/11/software.pdf>. [Último acceso: 02 Marzo 2020].
- [15 «EcuRed,» 04 Febrero 2012. [En línea]. Available:  
] [https://www.ecured.cu/Editor\\_de\\_texto](https://www.ecured.cu/Editor_de_texto). [Último acceso: 06 Marzo 2020].
- [16 J. P. P. y. M. Merino, «Definicion.de,» 2016. [En línea]. Available:  
] <https://definicion.de/compilador/>. [Último acceso: 06 Marzo 2020].
- [17 «enciclopedia,» 12 Abril 2009. [En línea]. Available:  
] [http://enciclopedia.us.es/index.php/Int%C3%A9rprete\\_inform%C3%A1tico](http://enciclopedia.us.es/index.php/Int%C3%A9rprete_inform%C3%A1tico). [Último acceso: 06 Marzo 2020].
- [18 «wikipedia,» 22 Octubre 2019. [En línea]. Available:  
] <https://es.wikipedia.org/wiki/Enlazador>. [Último acceso: 06 Marzo 2020].
- [19 «abrirllave.com,» [En línea]. Available: <https://www.abrirllave.com/diccionario-de-informatica/depurador-de-codigo.php>. [Último acceso: 06 Marzo 2020].
- [20 jadiaz, «miarduino,» 21 01 2016. [En línea]. Available: <https://bit.ly/2J0bmn0>.  
]
- [21 «pluselectric,» 21 09 2014. [En línea]. Available:  
] <https://pluselectric.wordpress.com/2014/09/21/arduino-uno-especificaciones-y-caracteristicas/>. [Último acceso: 13 Marzo 2020].
- [22 «Cracterísticas-Arduino.pdf,» 05 2013. [En línea]. Available:  
] <http://www3.gobiernodecanarias.org/medusa/ecoblog/ralvgon/files/2013/05/Caracter%C3%ADsticas-Arduino.pdf>. [Último acceso: 13 Marzo 2020].
- [23 «luis llamas,» [En línea]. Available: <https://www.luisllamas.es/temperatura-liquidados-arduino-ds18b20/>. [Último acceso: 09 Marzo 2020].
- [24 L. D. V. Hernández, «programa facil,» [En línea]. Available: <https://bit.ly/2Qx7Tft>.  
]
- [25 «Naylamp mechatronics,» [En línea]. Available:  
] <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hcsr04.html>. [Último acceso: 27 febrero 2020].

- [26 «koalab,» 24 abril 2017. [En línea]. Available:  
] <https://koalab.tech/aprende/componentes/sensor-ultrasonico-hc-sr04/>. [Último  
acceso: 27 febrero 2020].
- [27 «rambal automatizacion y robotica,» [En línea]. Available: <https://bit.ly/2VS3nyl>.  
]
- [28 «cicsa maxon,» [En línea]. Available: <https://bit.ly/2JVBFKn>.  
]
- [29 «wikipedia,» [En línea]. Available:  
] [https://es.wikipedia.org/wiki/Chispa\\_el%C3%A9ctrica](https://es.wikipedia.org/wiki/Chispa_el%C3%A9ctrica). [Último acceso: 26 febrero  
2020].
- [30 «C&DteCHNologia,» [En línea]. Available: [https://cdtecnologia.net/rele/1154-mini-  
\] rele-5v-10a-para-pcb.html](https://cdtecnologia.net/rele/1154-mini-rele-5v-10a-para-pcb.html). [Último acceso: 06 Marzo 2020].
- [31 M. B. David Cuartielles, «personales.upv.es,» [En línea]. Available:  
] [http://personales.upv.es/moimacar/download/arduino/arduino\\_rele.pdf](http://personales.upv.es/moimacar/download/arduino/arduino_rele.pdf).
- [32 jadiaz, «miarduino,» 11 05 2016. [En línea]. Available: <https://bit.ly/2X2nres>.  
]
- [33 J. P. P. y. María, «definicion,» 2009. [En línea]. Available:  
] <https://definicion.de/bluetooth/>. [Último acceso: 27 febrero 2020].
- [34 «La Comisión Federal de Comercio,» Septiembre 2011. [En línea]. Available:  
] <https://bit.ly/32HDgec>. [Último acceso: 02 Marzo 2020].
- [35 L. Díaz, «DIWO,» 14 Mayo 2015. [En línea]. Available: <https://bit.ly/2Th13OT>. [Último  
] acceso: 02 Marzo 2020].
- [36 07 2013. [En línea]. Available: <http://ito.mx/Loze>.  
]
- [37 A. inventor, «App inventor,» [En línea]. Available:  
] [http://ai2.appinventor.mit.edu/?locale=es\\_ES#4743703315480576](http://ai2.appinventor.mit.edu/?locale=es_ES#4743703315480576).
- [38 «curiosoando,» 13 abril 2015. [En línea]. Available: [https://curiosoando.com/que-es-  
\] una-valvula-solenoides](https://curiosoando.com/que-es-una-valvula-solenoides).
- [39 «arisa,» [En línea]. Available: <http://www.arisa.com.mx/molinos.html>. [Último acceso:  
] 06 Marzo 2020].

