



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE
TELECONTROL DE UNA SECADORA DE CACAO CON
INTERFAZ WEB MEDIANTE USO DE HARDWARE Y
SOFTWARE LIBRE”**

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN TELEMÁTICA

Presentado por:

WASHINGTON DAVID REYES ALVARADO

GUSTAVO DANIEL SANTIAGO ARIAS

GUAYAQUIL – ECUADOR

AÑO: 2015

AGRADECIMIENTO

Agradecemos en primer lugar a DIOS por darnos vida y sabiduría la cual nos ha permitido emprender esta carrera universitaria que no solo viene acompañada de conocimientos sino también de las experiencias tanto buenas como malas que no hubiéramos podido hacer frente sino fuera por los consejos brindados por parte de nuestros mentores académicos, amigos y familiares a los cuales agradecemos infinitamente, a ellos les dedicamos esta etapa de nuestras vidas.

DEDICATORIA

Dedico este trabajo a Dios, familiares, mentores académicos y espirituales, y amigos quienes han estado en todo tiempo a mi lado y han sido parte de esta hermosa experiencia académica.

Washington Reyes

Dedico este trabajo enteramente a mis padres quienes me han motivado siempre a seguir superando mis límites y su apoyo incondicional me ha llevado hasta aquí.

Gustavo Santiago

TRIBUNAL DE EVALUACIÓN

M. Sc. Marcos Millán Traverso

PROFESOR EVALUADOR

M. Sc. Miguel Molina

PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

.....
Washington David Reyes Alvarado

.....
Gustavo Daniel Santiago Arias

RESUMEN

El problema surge de la necesidad que poseen los agricultores que tienen sembríos de cacao, de poder encontrar un método de secado óptimo del grano de cacao, que además utilice la tecnología adecuada que le permita mejorar los tiempos de secado, y el grano no pierda sus propiedades de tal manera que los agricultores tras la venta de su producto logren obtener retribuciones económicas representativas.

En la actualidad existen diversas máquinas secadoras de cacao, pero que no cuentan la tecnología adecuada para poder optimizar sus procesos, para lo cual se ha planteado como solución a este problema el diseño e implementación de un prototipo de monitoreo que realice tareas de telecontrol y telemetría de una secadora de cacao existente para optimizar los procesos que realiza dicha maquina mediante una interfaz web, utilizando herramientas de software y hardware libre con el objetivo de reducir los costos de implementación.

Para la elaboración de este prototipo fue necesario utilizar la metodología del prototipado, que comienza con una planeación y análisis de los requerimientos que en este caso se obtuvo mediante una visita técnica y entrevista a una empresa que realiza secadoras de cacao, y una vez levantado los requerimientos se elaboró un cronograma de actividades que comenzó su ejecución con la investigación de la tecnología adecuada a utilizar y el diseño básico de lo que se habría de implementar en el prototipo, luego que se diseñó tanto los requerimientos mínimos de hardware como de software se presentó el diseño al fabricante de la secadora de cacao para su validación y se obtuvo una retroalimentación que sirvió para seguir mejorando el prototipo, una vez que se obtuvo un prototipo con los requerimientos básicos se comenzó a realizar pruebas en un maqueta diseñada para simular las entradas y salidas de señales que maneja el sistema y se obtuvo resultados favorables ya que el sistema respondió de manera adecuada donde se pudo demostrar que módulos que se implementaron en el sistema funcionan correctamente.

Cabe destacar que el prototipo que se logró elaborar tiene un costo bajo de implementación en comparación a productos de automatización existentes en el mercado y que manejan características similares, llegando a costar el doble de lo que costaría elaborar el prototipo que se plantea como solución.

ÍNDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iii
TRIBUNAL DE EVALUACIÓN	iv
DECLARACIÓN EXPRESA	v
RESUMEN	vi
ÍNDICE GENERAL.....	vii
CAPÍTULO 1	1
1. GENERALIDADES.....	1
1.1 Antecedentes	1
1.1.1 Procesamiento del cacao.....	1
1.1.2 Características de una secadora de cacao.....	4
1.2 Problemática	8
1.3 Justificación del Proyecto.....	8
1.4 Alcances y Limitaciones	9
1.5 Objetivos	10
1.5.1 Objetivo General	10
1.5.2 Objetivos Específicos	10
CAPÍTULO 2.....	11
2. SOLUCIÓN.....	11
2.1. Metodología	11
2.2. Algoritmo Propuesto.....	11
2.3. Diseño de Hardware	13
2.3.1. Controlador.....	14
2.3.2. Diseño de salidas en el hardware	18
2.3.3. Diseño de entradas en el hardware.....	21
2.4. Implementación del hardware	27
2.5. Diseño del software.....	28
2.5.1. Diseño del Modelo de Negocios.....	28

2.5.2. Diseño de la Vista	30
2.5.3. Diseño del Controlador.....	34
2.6. Implementación del software	39
2.7. Diseño de la Red de Comunicación.....	40
2.8. Costos de Implementación.....	45
CAPÍTULO 3.....	47
3. ANALISIS DE PRUEBAS Y RESULTADOS.....	47
3.1. Escenario 1	47
3.1.1 Resultados	48
3.2. Escenario 2.....	48
3.2.1. Resultados	48
3.3. Escenario 3.....	49
3.3.1. Resultados	49
CONCLUSIONES Y RECOMENDACIONES	51
BIBLIOGRAFÍA.....	53
ANEXOS.....	54

CAPÍTULO 1

1. GENERALIDADES.

1.1 Antecedentes

Ecuador es un país que tiene un nivel muy alto de producción de cacao, el cual es utilizado como materia prima para la fabricación de diferentes productos como son el chocolate o la pasta de cacao, sin embargo la gran mayoría de agricultores que cosechan el cacao no cuentan con la maquinaria necesaria para procesarlo sino que en su gran mayoría lo venden en grano para que sea procesado.

El valor al que lo venden depende del peso y del grado de humedad del cacao ya que si está muy húmedo, el comprador deberá realizar la labor de secado hasta que el producto pueda ser procesado, es por esta razón que muchos agricultores que trabajan con el cacao se preocupan por secarlo adecuadamente para que sus ganancias no se vean afectadas.

1.1.1 Procesamiento del cacao.

La preparación del cacao como materia prima para la industrialización incluye un proceso que inicia desde la cosecha de las mazorcas maduras, sigue con la extracción de la pulpa o almendra, la fermentación y culmina con el secado del grano, para después comenzar otro proceso de tratado según sea el derivado que se quería elaborar.

El fin de esta primera etapa es convertir la materia prima (grano de cacao) en un producto comercial de calidad, y para esto se deben considerar variables como el tipo genético, el medio donde se encuentra el cultivo, el método de fermentación y secado.

La cosecha se realiza cuando la mazorca ha llegado a su madurez, lo que se aprecia por el cambio de coloración de la misma, durante la cosecha se escogen las mazorcas sanas, desechando las que presentan problemas (enfermas), En

Ecuador la floración principal se da en julio y junio, y una segunda floración en septiembre y octubre, el periodo de maduración oscila entre los 4 a 6 meses, según la altura sobre el nivel del mar y la temperatura del lugar, así la cosecha más fuerte se da en los meses de octubre, noviembre y diciembre y la segunda pero en menor cantidad durante los meses de marzo y abril.

- El proceso de extracción del grano generalmente se realiza de manera manual, aunque actualmente se están creando máquinas para de extracción de granos de cacao, sin embargo no son recomendadas por le ineficiencia que presentan y el trato del grano debe ser de sumo cuidado para no lastimarlo, luego de la extracción el cacao está listo para la fermentación.
- El proceso de fermentación conocida también como cura, preparación, cocinado, etc. Es el proceso al que es sometido las almendras frescas donde se mata el embrión y permite a las almendras adquirir su particular calidad, haciendo de este proceso el más importante para que el sabor y aroma sea el adecuado por ejemplo para la elaboración de chocolate de calidad, aunque el proceso de fermentación se da en diferentes maneras, el más común y más empleado es en sacos, en cajas, o haciendo montones del grano en grandes tendales, luego que ha sido fermentado el cacao está listo para ser secado.
- El proceso de secado, es el proceso durante el cual las almendras terminan de perder el exceso de humedad que contienen y están listas para ser vendidas o procesadas en los siguientes procesos de elaboración de derivados, durante este proceso se producen cambios en el color, apareciendo exteriormente con un color típico marrón, el secado puede ser de manera natural o artificial mediante el uso de secadoras.

El secado natural se lo realiza aprovechando el calor del sol, para lo cual se “tienden” los granos de cacao sobre tendales de caña

o cemento donde se lo mantiene a la intemperie recibiendo los rayos del sol, este proceso puede durar aproximadamente 3 días, según sean las condiciones climáticas, durante el periodo que el cacao permanece expuesto al sol se debe rastrillar el mismo, es decir remover el cacao de tal manera que los granos puedan recibir el sol íntegramente, y no parcialmente, esta actividad debe ser realizada varias veces al día.



Figura 1.1: Foto de proceso de secado de Cacao manual. [1]

En la Figura 1.1 se observa como una persona realiza la tarea de secado en un tendal o superficie realizando un esfuerzo muy grande mediante instrumentos de maderas llamadas “rastrillos” que le ayudan a remover el grano para que puedan secarse de manera regular.

El secado artificial se efectúa mediante secadoras mecánicas que por lo general se utilizan en lugares no prestan las condiciones climáticas para un secado natural óptimo o por la falta de tendales, el principio de funcionamiento de estas máquinas es hacer que fluya aire seco y caliente por las almendras de cacao. Existen diferentes tipos de mecanismos con los que se pueden realizar este tipo de secado, siendo algunos

más efectivos que otros, actualmente algunas de estas secadoras han encontrado mecanismos que permiten que el secado que normalmente duraba días en un proceso natural, ahora pueda durar pocas horas, dependiendo de la cantidad de cacao que se pretenda secar. [2]



Figura 1.2: Foto de proceso de secado de Cacao artificial. [3]

En la Figura 1.2 se puede observar un maquina secadora de cacao tipo industrial que cuenta con removedor de granos lo que hace que el cacao pueda secarse en menor tiempo y de manera homogénea.

1.1.2 Características de una secadora de cacao.

La secadora de cacao que vamos a utilizar como referencia para adaptar nuestro sistema, actualmente se encuentra instalada en la ciudad de Milagro provincia del Guayas, Ecuador, y posee una estructura tipo bandeja circular, el encendido es manual y a través de una antorcha permite que la recámara de combustión encienda.

Hace uso de un Variador de Frecuencia para controlar los ciclos para remover el cacao utilizando unos Timers analógicos que permitirá al motor activarse y remover el cacao cada cierto tiempo.

La secadora de cacao opera haciendo uso de diferentes dispositivos, los cuales vamos a describir a continuación incluyendo las funciones que cumple cada uno de ellos:



Figura 1.3: Recamara de calor y plato de acero de secadora.

- En la Figura 1.3 se muestra la recamara de calor, donde fluye el aire caliente, esta es de tipo cilíndrica y está recubierta de cemento, también se muestra en la parte superior el plato de acero inoxidable, y es donde se colocan los granos de cacao que se van a secar.



Figura 1.4: Motor que moviliza las aspas.

- En la Figura 1.4 se muestra el motor, que moviliza las aspas ubicadas encima del plato de acero para darle rotación a los granos y obtener un secado óptimo.



Figura 1.5: Panel de control.

- En la Figura 1.5 se muestra el panel de control, que contiene un variador de frecuencia, timers y botones de encendido y apagado de la máquina.



Figura 1.6: Ventilador Industrial.

- En la Figura 1.6 se muestra un ventilador industrial, que está conectado a un PLC el cual varía su velocidad según el flujo de aire requerido para aumentar o disminuir la temperatura en la recámara.



Figura 1.7: Panel del PLC y Electroválvulas.

En la Figura 1.7 se muestra el panel del PLC, que está conectado al ventilador, sensores de temperatura dentro de la recámara de calor y con electroválvulas que dan paso al gas, según la programación del PLC.

1.2 Problemática

El problema surge de la necesidad que poseen los agricultores que tienen sembríos de cacao, de poder encontrar un método de secado óptimo del grano de cacao y conocer las temperaturas a las que estuvo sometido el cacao y el grado de humedad resultante al finalizar el secado para conocer si el cacao se ha secado en exceso o si necesita secarse más.

El proceso de secado normalmente toma mucho tiempo, y debido a que se debe tener un control en todo momento del estado del cacao y a que debe ser removido constantemente, el agricultor debe detener sus actividades enteramente para poder monitorear el proceso él mismo. Esto afecta la productividad del agricultor ya que prácticamente lo obliga a perder un día de trabajo.

1.3 Justificación del Proyecto

Nuestro proyecto tiene la intención de cubrir las necesidades antes mencionadas brindándole la oportunidad al agricultor de secar apropiadamente su cacao utilizando una secadora inteligente y de bajo costo para permitir que el agricultor pueda incrementar sus ganancias puesto que con cada cosecha el agricultor pierde de un 10 a un 20% de las ganancias las cuales pasan a manos del comprador.

También es importante mencionar que si se llevaran a cabo proyectos similares automatizando las máquinas que procesan el cacao utilizando hardware y software libre, sería posible que el cacao sea procesado por el propio agricultor con lo cual se reduciría la tasa de exportación de cacao como materia prima y aumentaría la tasa de exportación de productos elaborados a base del cacao, reflejándose esto en un incremento en el capital que ingresa al país dirigido al sector agrícola potenciado por tecnología de bajo costo y al alcance.

El sistema permitirá la automatización del proceso de encendido de la secadora y secado del cacao, monitoreando el sistema constantemente

y mostrando reportes de los procesos de secado realizados con anterioridad que ayudarán al agricultor a mejorar el proceso de secado brindándole la oportunidad de controlar y monitorear remotamente el secado de su producto y permitiéndole continuar con sus actividades laborales mientras el sistema automáticamente deja el cacao listo para ser recogido y llevado a vender.

1.4 Alcances y Limitaciones

El sistema tendrá ciertos alcances y limitaciones con respecto a la manera en que se seca el cacao, el sistema permite configurar procesos de secado y programarlos durante un tiempo determinado, pero no es capaz de calibrar la temperatura de secado en base a la humedad del cacao lo cual sería lo ideal.

Configurar un sistema que varíe la temperatura en base a la humedad del cacao con el hardware y el software que hemos diseñado no representa en realidad ningún problema, la verdadera dificultad reside en **conocer la humedad del grano de cacao**.

La humedad en un grano de cacao no puede ser medida con facilidad porque es necesario partir el grano para conocer la humedad dentro de la almendra, por lo tanto un sensor de humedad entregará un grado de humedad erróneo ya que está midiendo la humedad externa del grano de cacao.

Por esta razón hemos decidido replicar la lógica tradicional de secado de cacao, que consiste en, una vez conocido el peso del cacao, dejarlo secar por determinado número de horas, obviamente mientras más peso tiene el cacao, mayor número de horas necesitará.

Y con la ayuda del sistema de reportes, es posible conocer en determinado proceso, el tiempo de secado, el peso del cacao y la temperatura a la que estuvo expuesto y es decisión del usuario determinar si el grado de humedad fue aceptable al finalizar el proceso,

o si para futuros procesos necesita aumentar o disminuir el tiempo de secado.

Otro aspecto importante es que el usuario tiene la facilidad de detener el proceso en todo momento si considera que el cacao ya está seco, o por el contrario de iniciar otro proceso inmediatamente si no está lo suficientemente seco.

El sistema embebido en la tarjeta BeagleBone Black se inicia automáticamente al encender la tarjeta, por lo tanto si ocurriera algún corte de energía, el usuario no debe preocuparse por configurar o iniciar algún proceso en la tarjeta puesto que cuando la energía regrese automáticamente entrará en funcionamiento.

1.5 Objetivos

1.5.1 Objetivo General

Desarrollar e implementar un prototipo tecnológico de bajo costo que permita realizar tareas de telemetría y telecontrol para optimizar los procesos de una secadora de cacao.

1.5.2 Objetivos Específicos

- Usar hardware y software libre en el diseño del prototipo para la reducción de costes de implementación.
- Crear una interfaz de usuario minimalista y adaptable al dispositivo de acceso para que el usuario pueda operar el sistema de manera sencilla e intuitiva.
- Implementar seguridades en el ingreso al sistema mediante autenticación de usuario.
- Brindar en la interfaz web una sección de reportes de todos los procesos de secado y datos asociados a ellos, mediante la implementación de una base de datos.

CAPÍTULO 2

2. SOLUCIÓN.

En la actualidad existen diversas máquinas secadoras de cacao, pero que no cuentan con la tecnología adecuada para poder optimizar sus procesos, para lo cual se ha planteado como solución a este problema el diseño e implementación de un prototipo de monitoreo que realice tareas de telecontrol y telemetría de una secadora de cacao existente para optimizar los procesos que realiza dicha maquina mediante una interfaz web, utilizando herramientas de software y hardware libre con el objetivo de reducir los costos de implementación.

2.1. Metodología

La metodología que se utiliza en este proyecto es la del prototipado, y se comienza con una planeación y análisis de los requerimientos que en este caso se obtuvo mediando una visita técnica y entrevista a una empresa que realiza secadoras de cacao y tiene procesos definidos para el secado, una vez levantado los requerimientos se elaboró un cronograma de actividades con línea de tiempo definida y se comenzó su ejecución con la investigación de tecnología a utilizar y el diseño básico de lo que se habría de implementar en el prototipo, luego que se diseñó tanto los requerimientos mínimos de hardware como de software se presentó el diseño al fabricante de la secadora de cacao para su validación y se obtuvo una retroalimentación que sirvió para seguir mejorando el prototipo hasta que estuvo apto para realizar las pruebas en una maqueta de simulación.

2.2. Algoritmo Propuesto

Para comenzar con el diseño del hardware y del software se tiene que definir el algoritmo del sistema para saber cuáles son las operaciones a realizar.

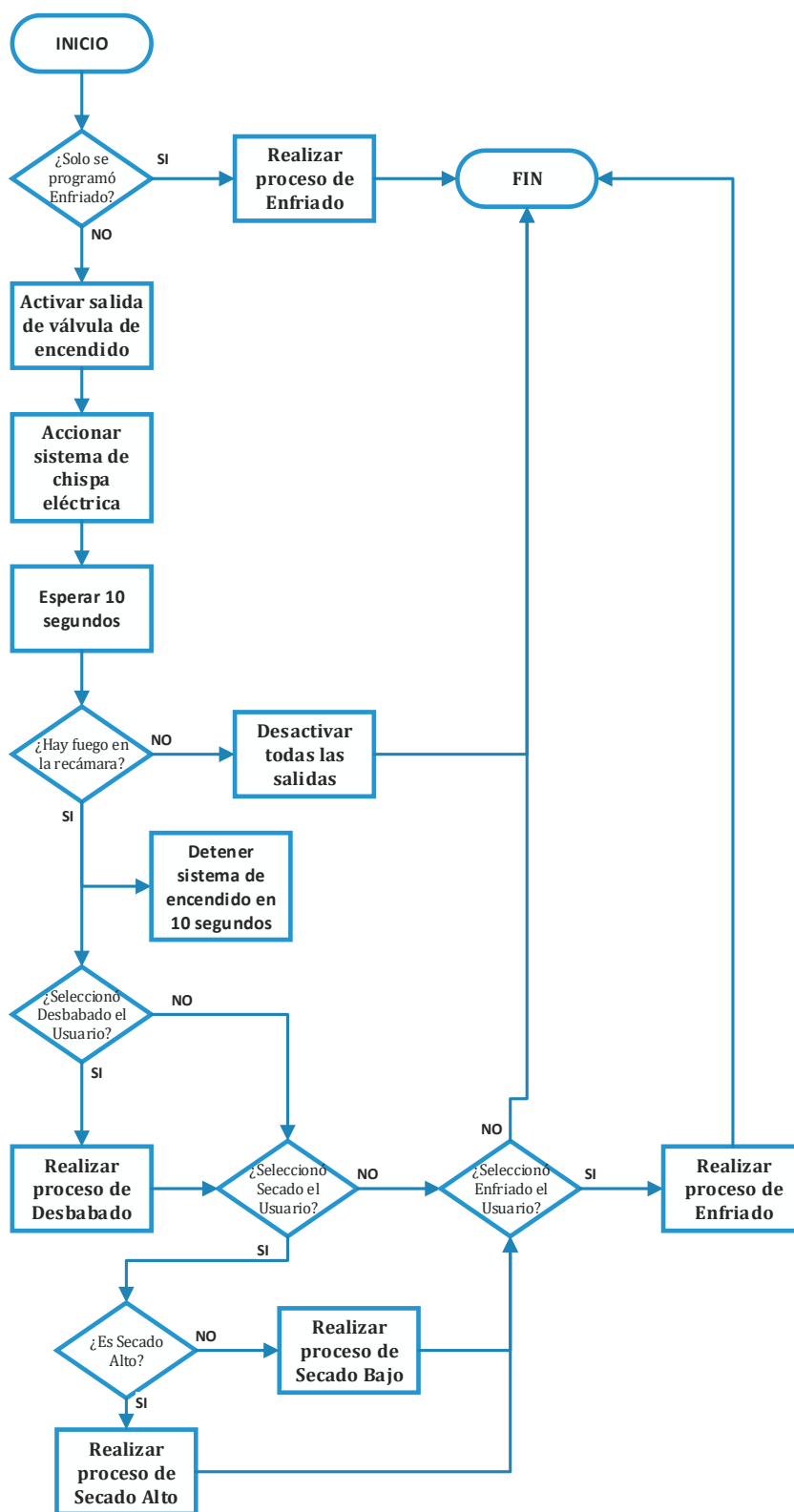


Figura 2.1: Algoritmo general del sistema.

La Figura 2.1 muestra el algoritmo general de cómo funcionará la maquina secadora de cacao con la implementación del sistema de monitoreo.

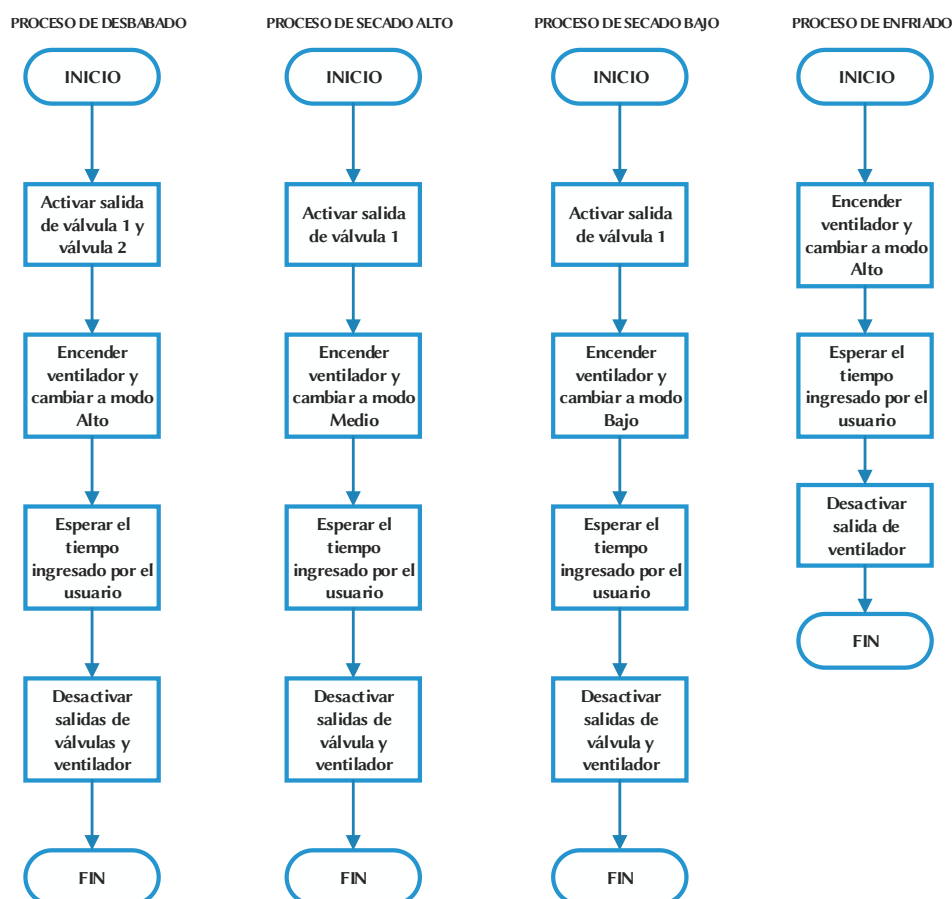


Figura 2.2: Algoritmo de procesos individuales del sistema.

La Figura 2.2 muestra los algoritmos por procesos, que se pueden programar desde el sistema de monitoreo.

2.3. Diseño de Hardware

El sistema completo de la secadora consta de 3 electroválvulas para el paso de gas, un ventilador con 3 niveles de velocidad, un sistema de encendido automático mediante arco eléctrico, un sensor para detectar la cantidad de luz en el túnel donde se produce la llama, un sensor de

temperatura y un sistema temporizador que activa un motor cada cierto tiempo para remover el cacao a través de unas aspas colocadas en la secadora.

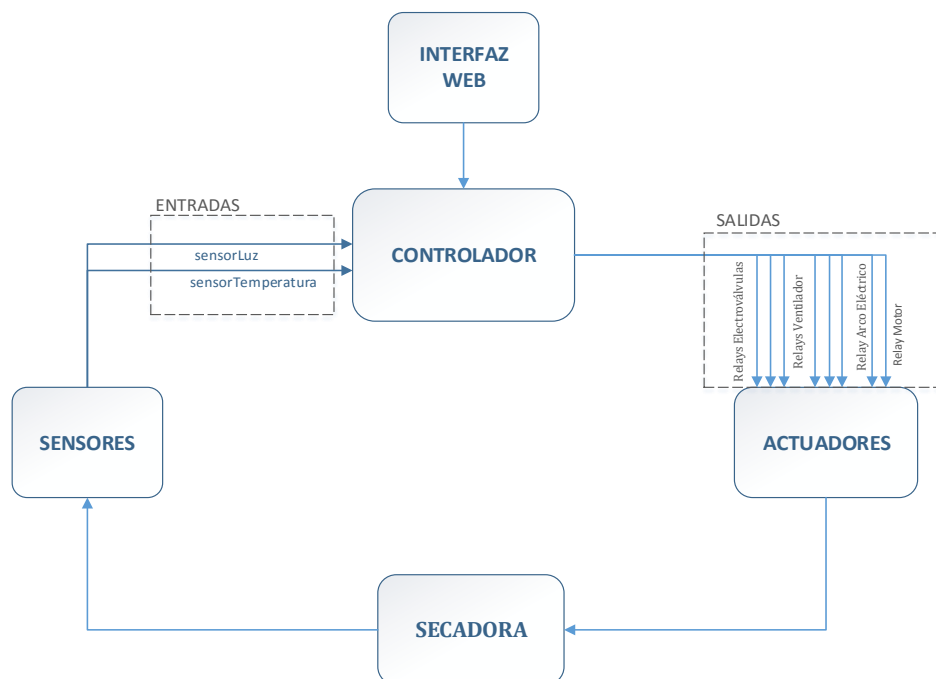


Figura 2.3: Diagrama de bloques del sistema.

La Figura 2.3 muestra un diagrama de bloques del sistema donde se puede observar claramente las entradas y salidas del controlador.

Para comunicar al controlador con los actuadores y con los sensores necesitaremos de un circuito externo que controle los actuadores por medio de relés (pre-actuadores), y también un circuito de acondicionamiento para las señales que reciben los sensores.

2.3.1. Controlador

Actualmente en el mercado existe una gran variedad de estos dispositivos, junto con una documentación bastante extensa sobre su funcionamiento y aplicaciones, así que hemos recopilado algunas de las características más importantes de las tarjetas SBC más conocidas en el mercado.

	Arduino Yun	BeagleBone Black	Intel Galileo	Raspberry Pi
VELOCIDAD DE CPU	400 MHz	1 GHz	400 MHz	700 MHz
MEMORIA RAM	64 MB	512 MB	256 MB	512 MB (modelo B)
ALMACENAMIENTO INTERNO	16 MB	4 GB (rev. C)	8 MB	Ninguno
GPU	Ninguno	PowerVR SGX530	Ninguno	Broadcom VideoCore IV
PRECIO APROXIMADO (USA)	\$75	\$55	\$80	\$35
PINES DIGITALES I/O	20	65	14	17
ENTRADAS ANALÓGICAS	12 (10 bits)	7 (12 bits)	6 (12 bits)	Ninguno
SALIDAS PWM	7	8	6	1
VOLTAJE I/O	5V	3.3V	3.3 o 5V	3.3V

Tabla 1: Comparativa entre diferentes SBC

Con los datos comparativos adjuntos en la Tabla 1 solo cabe destacar que Raspberry Pi y BeagleBone Black son las únicas tarjetas que poseen soporte para una gran variedad de sistemas operativos en distribuciones conocidas y de fácil uso como Ubuntu, Debian, Android, en caso que no se desee utilizar el sistema operativo de fábrica puesto que a veces éste posee funciones muy limitadas.

También podemos notar los diferentes precios de cada una de las SBC que van a influir también en la decisión final.

Hay que prestar mucha atención a la cantidad de pines analógicos que puede manejar cada tarjeta también, ya que nuestro proyecto incluirá sensores y convendría utilizar una tarjeta que posea pines analógicos para procesar las señales sin necesidad de utilizar un Convertidor Analógico / Digital.

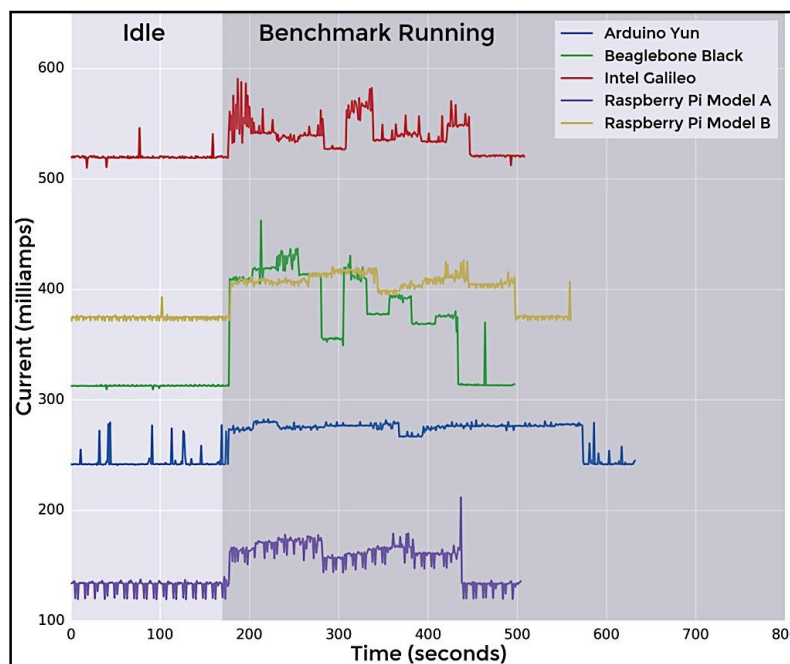


Figura 2.4: Consumo de corriente de las SBC [4]

Observando la Figura 2.4 que indica la corriente (proporcional a la potencia) que consume cada SBC y en conjunto con la Tabla 1, hemos descartado las opciones de Intel Galileo por su alto consumo de energía y costo elevado y Arduino Yun por sus limitaciones de espacio (16 MB disponibles para todo el S.O.) que impiden trabajar con compiladores de código internamente.

De tal manera que hemos decidido analizar un poco más a fondo las tarjetas Raspberry Pi y BeagleBone Black destacando las ventajas y desventajas de cada una de ellas.

RESUMEN DE LAS VENTAJAS Y DESVENTAJAS DE RASPBERRY PI

Ventajas

- Bajo Costo (\$35 en USA).
- Bajo consumo de energía.

- Variedad de sistemas operativos para trabajar.
- Librerías fáciles de usar para trabajar con pines I/O.
- Tiene gran cantidad de tutoriales y guías disponibles en línea debido a ser la tarjeta más popular entre la comunidad de Internet.

Desventajas

- No posee entradas analógicas.
- No posee almacenamiento interno, depende de una tarjeta SD.
- Velocidad de procesamiento menor (700 MHz).

RESUMEN DE LAS VENTAJAS Y DESVENTAJAS DE BEAGLEBONE BLACK.

Ventajas

- Tiene 65 entradas digitales.
- Mayor velocidad de procesamiento (1 GHz).
- Posee almacenamiento interno de 4 GB.
- Tiene 7 entradas analógicas con una resolución de 12 bits.
- Provee un IDE basado en web, y una librería llamada BoneScript que permiten un fácil manejo de los pines GPIO.
- Algunos de los pines pueden ser configurados para múltiples funciones diferentes como Timers o PWM.
- Soporta Debian, Ubuntu, Android, ArchLinux entre otros.

Desventajas

- Costo medio (\$55 en USA)
- No existe mucha documentación y guías como el caso de Raspberry dado que tiene relativamente poco tiempo en el mercado.

Debido a estas consideraciones, hemos decidido trabajar con la tarjeta BeagleBone Black la cual parece satisfacer nuestras necesidades al permitirnos implementar un servidor web, servidor de bases de datos, y un controlador utilizando sus entradas analógicas para sensar los datos externos sin necesidad de un convertidor Analógico / Digital, y utilizando sus salidas digitales para controlar un circuito externo que activará los actuadores del sistema, todo esto a un costo relativamente bajo y con gran potencial debido a la facilidad de programación de los pines GPIO en lenguajes como Python, JavaScript o C++ y debido a Cloud9, el IDE que viene incorporado que permitirá hacer depuraciones al sistema para ver cómo responden las salidas y entradas al momento de ejecutar el código fuente del controlador.

2.3.2. Diseño de salidas en el hardware

- **Electroválvulas**

Para controlar si una electroválvula se enciende o se apaga mediante los pines de la BeagleBone debemos tomar en consideración cuanta corriente es capaz de suministrar un pin de la BeagleBone, dicha información se encuentra en la hoja de datos del procesador e indica que la mayoría de los pines pueden manejar hasta 6mA de corriente con excepción de unos pocos que manejan hasta 4mA [5].

Esta corriente no es suficiente para nuestros requerimientos por lo cual es necesario el uso de un transistor que trabaje como conmutador y no como amplificador.

La salida del GPIO va a provocar un cambio de estado en el transistor (entre corte y saturación), que nos permitirá activar un relé usando una fuente de voltaje externa.

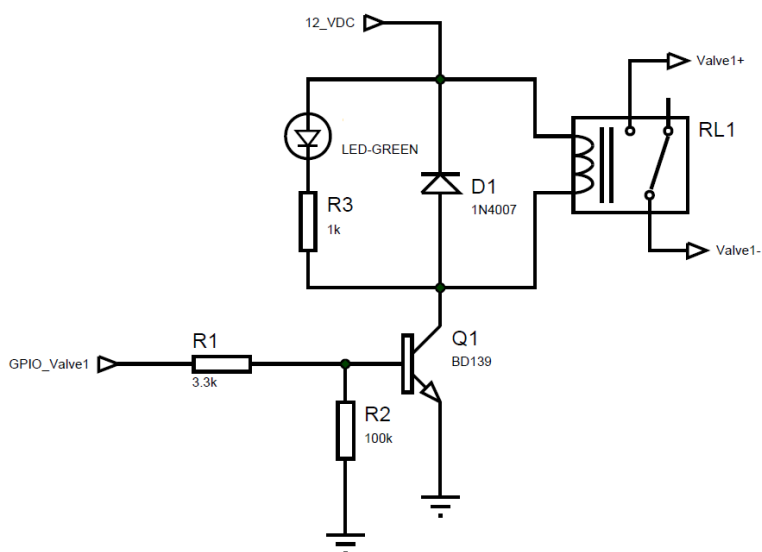


Figura 2.5: Diagrama del circuito para electroválvula.

En la Figura 2.5 podemos observar el circuito utilizado para controlar las válvulas, explicando a breves rasgos al activarse el GPIO entregará 3.3v el cual corresponde a un estado de ALTO digital para la BeagleBone Black, **R1 controla la cantidad de corriente** entregada a la base del transistor la cual es alrededor de 780 μA , **R2 es una resistencia a tierra** para evitar que el transistor active el relé cuando la BeagleBone está cargando la configuración de pines y dichos pines aún no están configurados como entrada/salida sino en alta impedancia, en ese caso gracias

a R2 la base del transistor se conecta a tierra asegurando que el circuito esté desactivado.

El relé que hemos utilizado funciona con 12V con una bobina de 400 ohms, hemos puesto el **diodo D1 como protección** debido a que un inductor se comporta de tal manera que se opone a los cambios bruscos de corriente entregando un voltaje proporcional al cambio de corriente, esto se complica al momento de desactivar el relé el cual genera un voltaje que podría dañar nuestro transistor o a la larga los contactos del relé.

Un LED nos indica el estado del relé y la electroválvula estará conectada con su respectiva fuente entre los contactos **normalmente abierto y común** del relé, el resultado será que cuando la salida está en ALTO, el relé se activará uniendo los contactos y activando la electroválvula y cuando la salida está en BAJO el relé se desactiva y los contactos se separan desactivando la electroválvula.

- **Ventilador**

Para controlar las diferentes velocidades del ventilador usaremos el mismo circuito utilizado en las electroválvulas, conectando los contactos normalmente abiertos de cada relé al circuito variador de velocidad interno del ventilador de tal manera que vamos seleccionando una resistencia diferente para obtener una velocidad diferente, de esta manera conseguiremos tres velocidades, alta media y baja usando tres circuitos similares al de la Figura 2.5.

- **Encendido mediante arco eléctrico**

El mismo circuito presentado en la Figura 2.5 será utilizado para activar el arco eléctrico usando los contactos del relé

como switch para activar el arco eléctrico o desactivarlo, la chispa puede ser producida por una pequeña bujía.

- **Motor**

El motor en la secadora real utiliza un Variador de Frecuencia para establecer los períodos en los cuales el motor se activa empujando las aspas para remover el cacao, pero para efectos del prototipo hemos utilizado un motor DC con un sistema de engranajes para reducir la velocidad del mismo y para poder activarlo cada cierto tiempo utilizaremos otro relé de tal manera que el mismo circuito pueda ser utilizado en la secadora real ya que el Variador de Frecuencia necesita una señal de cortocircuito para accionarse y dicha señal puede salir del mismo relé.

2.3.3. Diseño de entradas en el hardware

- **Sensor de luz**

La función de este sensor es **determinar si el encendido automático fue exitoso y monitorear en todo momento si la recámara de combustión sigue encendida**, en caso de no haber encendido o de apagarse la recámara por algún motivo será necesario hacer una parada de emergencia en el sistema, cerrar las válvulas de paso de gas, desactivar el ventilador y el motor y reflejar lo sucedido en el reporte indicando en qué momento se detuvo la operación.

Dichos eventos podrían ocurrir si se terminó el gas de los tanques, el ventilador se averió y no envía aire a la recámara de combustión o si sucedió algún inconveniente al momento del encendido.

El sensor utilizado es una fotocelda o LDR (Light-Dependent Resistor) y mediante un divisor de voltaje envía una señal

analógica de la luminosidad existente dentro de la cámara de combustión.

¿Por qué usamos una fotocelda y no un fototransistor?

Los fototransistores tienen varias ventajas puesto que detectan rápidamente las variaciones de luz en comparación a una fotocelda, y son muy utilizados en circuitos detectores de línea o detectores de velocidad de rotación de un motor debido a su característica de respuesta rápida, pero nosotros hemos utilizado una fotocelda por diferentes motivos:

- ✓ Su amplia respuesta en frecuencia, es decir es sensible a diferentes longitudes de onda prácticamente cubriendo todo el espectro visible.
- ✓ Bajo costo en comparación a un fototransistor.
- ✓ Mejor desempeño frente a un sistema sometido a vibraciones provocadas por el motor que remueve el cacao en la secadora.
- ✓ Mejor desempeño en altas temperaturas.
- ✓ Mayor facilidad para ser colocada alrededor de la válvula de encendido.

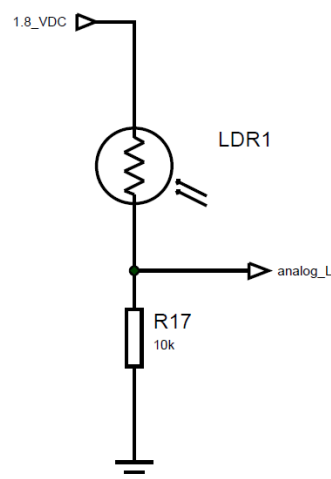


Figura 2.6: Circuito correspondiente al sensor de Luz.

En la Figura 2.6 observamos el circuito que utiliza un voltaje de referencia tomado del pin P9_32 de la BeagleBone el cual suministra 1.8V, el circuito no es más que un divisor de voltaje entre la foto-resistencia y una resistencia fija de 10K ohms, la salida va conectada directamente a un pin analógico de la BeagleBone y al estar en completa oscuridad dentro de la cámara de combustión, la resistencia de la LDR es tan alta que prácticamente “abre” el circuito conectando la salida a tierra, y cuando la cámara se encienda, la luz provocada por la llama reducirá la resistencia de la LDR provocando una división de voltaje que será nuestra entrada y se procesará para indicar que la ignición fue exitosa y monitorear el nivel de intensidad de luz aproximado.

- **Sensor de Temperatura**

Existen diferentes sensores de temperatura como son las termocuplas, RTD, termistores, pero hay que considerar varios aspectos para poder elegir el sensor adecuado para medir la temperatura dentro de la secadora.

- **Termocupla**

Una termocupla se conforma de la unión de dos metales diferentes que producen una señal de voltaje muy pequeña en función de la temperatura a la que está sometido el termopar, puede medir un rango de temperaturas bastante amplio, pero existen muchos problemas al momento de amplificar la señal debido a que se creará otra “termocupla” entre los contactos del amplificador y el cable de la termocupla dado que son dos metales distintos, y como esta señal tan pequeña viaja por un cable de aproximadamente dos metros de largo que se comporta como una antena, es susceptible al ruido eléctrico que produce el motor que remueve el cacao, por lo tanto las mediciones resultarán

erróneas utilizando este tipo de sensor a menos que se tomen estas consideraciones para el diseño del circuito amplificador y se compensen los valores de voltaje.

Los termistores por otro lado son de bajo costo, tienen una buena sensibilidad al cambio de temperatura, buen desempeño frente a vibraciones mecánicas, no generan un voltaje como lo hace la termocupla sino que su resistencia varía con respecto a la variación de temperatura, sin embargo esta respuesta "Resistencia Vs Temperatura" no es lineal lo cual hace difícil la medición exacta del valor de temperatura.

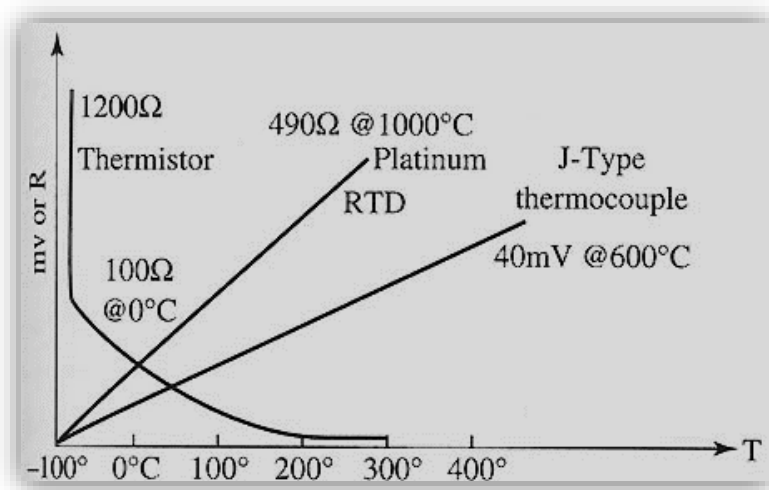


Figura 2.7: Linealidad entre diferentes sensores.

En la Figura 2.7 se muestra la comparación entre los tres sensores que estamos analizando y su respuesta frente a la variación de temperatura, aquí se puede observar la diferencia entre un termistor y una RTD en términos de linealidad.

- **RTD**

Un RTD (Resistance Temperature Detector) es un sensor que varía su resistencia dependiendo de la temperatura, provee una precisión de $\pm 0.8^\circ\text{C}$ alrededor de los 100°C que es la zona donde trabaja la secadora de cacao, puede detectar temperaturas hasta los $300\text{-}400^\circ\text{C}$ y dado que su respuesta es lineal, la medición resulta bastante sencilla, debido a que es un sensor que varía su resistencia y a su respuesta lineal en el rango de temperaturas que nos interesa (0° a 120°C) hemos elegido el RTD PT100 el cual tiene la característica que a 0°C tiene una resistencia de 100 ohms.

Pero como este sensor varía su resistencia debemos utilizar un circuito que nos permita obtener un voltaje, sin olvidar que los pines analógicos de la BeagleBone solo pueden soportar hasta 1.8 V.

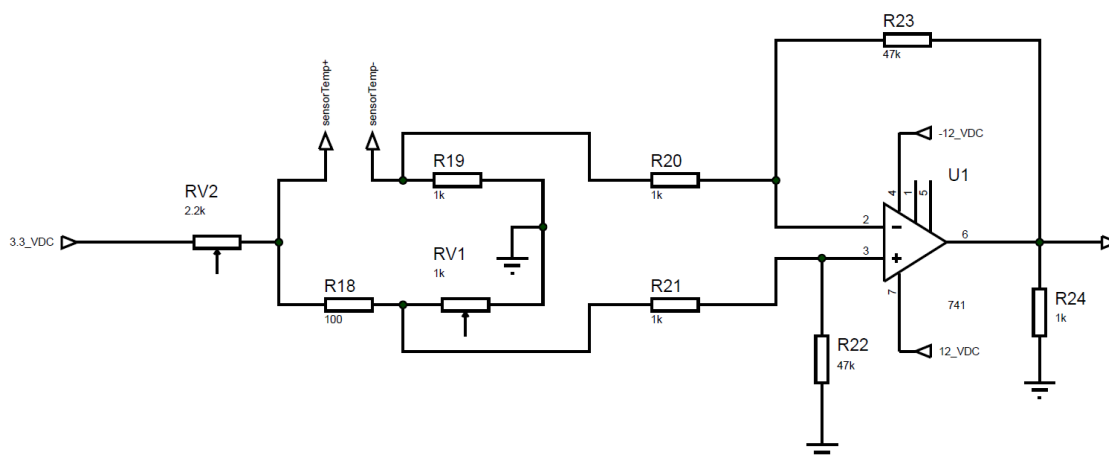


Figura 2.8: Circuito correspondiente al sensor de Temperatura

En la Figura 2.8 se muestra el circuito utilizado para medir la temperatura utilizando el sensor RTD PT100.

El circuito básicamente se conforma de un puente de Wheatstone donde nuestro sensor estará conectado como una de las resistencias del puente, la diferencia de potencial entre los extremos del puente **debe ser cero a 0°C**, para lograr esto R19 debe ser igual a RV1 y **R18 debe ser igual a la resistencia del sensor a 0°C** la cual es conocida y es 100 ohms, RV2 es un potenciómetro que **controla el paso de corriente al puente** ya que una corriente de más de 1mA podría dañar el sensor o provocar que se caliente producto de la energía disipada por el sensor alterando la medición de la temperatura, actualmente el potenciómetro en 2.2K ohms permite una corriente de 600uA a través del puente, RV1 también es un potenciómetro básicamente utilizado para calibrar el puente en caso de que las resistencias no sean iguales puesto que cuando se compra una resistencia, su valor tiene un nivel de tolerancia ya sea del 10%, 5%, 1% pero no es el valor exacto.

La variación en la resistencia del sensor es tan pequeña que los cambios de voltaje también son pequeños por lo tanto utilizamos un amplificador cuya ganancia es 47 para poder aumentar el voltaje de tal manera que podamos diferenciarlo pero sin exceder 1.8V ya que podría dañar la tarjeta.

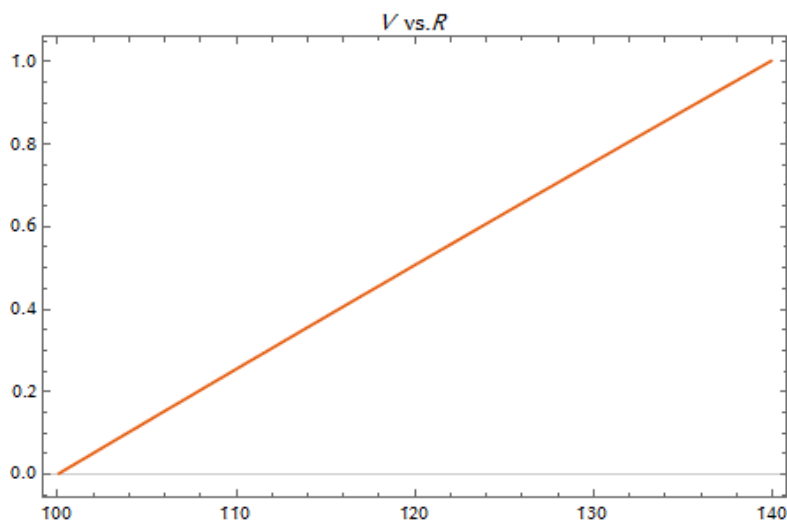


Figura 2.9: Función Voltaje Vs. Resistencia

La función presente en la Figura 2.9 indica cual es el valor del voltaje de salida cuando el sensor tiene una determinada resistencia (la cual varía de 100 a 140 ohms entre 0° y 104°C).

El circuito de medición de temperatura no puede exceder el voltaje crítico (1.8V) en las condiciones en que se encuentra debido a que la secadora opera hasta una temperatura de 100°C como máximo y se necesitaría de una temperatura alrededor de los 200°C para exceder dicho voltaje.

La linealidad presentada en la Figura 3.5 hace muy sencilla la programación a la hora de recibir los datos de voltaje del sensor de temperatura y mostrarlos a través de la interfaz web.

2.4. Implementación del hardware

Segun el diseño planteado en la sección anterior se ha diseñado una placa electrónica, para ser usada en la secadora de cacao, la cual cuenta con todas las entradas y salidas como fue anteriormente descrito.

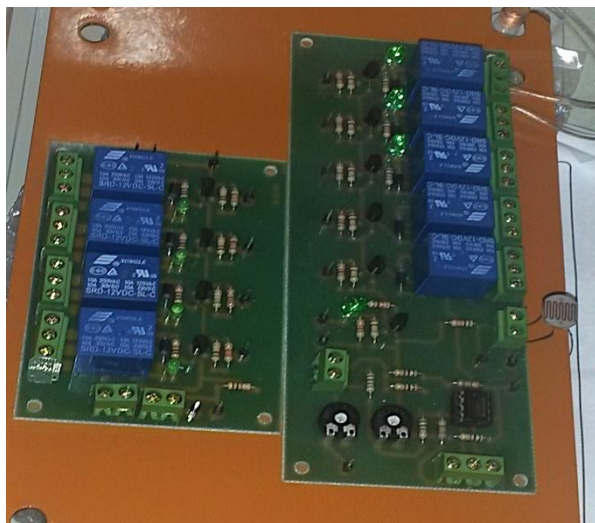


Figura 2.10: Implementación de placa electrónica.

Esta placa electrónica mostrada en la Figura 2.10 va directamente conectada con la BeagleBone Black, en sus respectivos pines GPIO definidos en el controlador.

2.5. Diseño del software

Una vez establecidos los componentes que vamos a controlar y los sensores a utilizar, y definido el diseño de los circuitos que van a actuar sobre estos componentes, necesitamos un software capaz de controlarlos bajo las condiciones que rigen el proceso de secado, así como también un modelo de Bases de Datos acorde al sistema que vamos a implementar y debemos definir cómo van a relacionarse entre sí el Modelo, la Vista y el Controlador.

2.5.1. Diseño del Modelo de Negocios

En nuestro modelo de negocios se decidió utilizar un SGBDR que sea open source y el que mejor se ajustaba a nuestros requerimientos fue MySQL debido a su flexibilidad, facilidad de manejo y sobre todo porque no demanda muchos recursos del sistema.

Se crearon tablas para representar los procesos de secado, la programación, el usuario del sistema y el lote de cacao a secar, y se las relacionó con las tablas correspondientes para asociar correctamente estos datos y cada vez que se programe un proceso se guarde un registro del mismo y se pueda obtener una información detallada en el módulo de reportes.

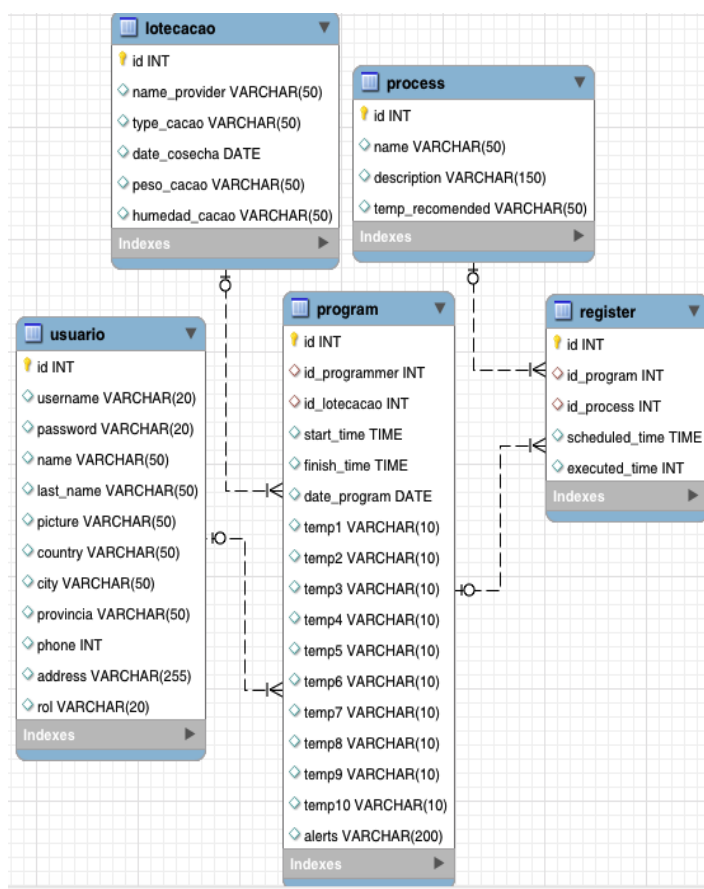


Figura 2.11: Modelo relacional de base de datos.

En la Figura 2.11 se presenta el esquema de bases de datos, que hemos planteado para los requerimientos de nuestra aplicación web.

2.5.2. Diseño de la Vista

La maquetación de la vista, es decir del front-end o interfaz de usuario fue realizada utilizando HTML5 para seccionar lo diferentes elementos que se presentarán al usuario, el diseño y estilos de estos elementos fueron descritos con CSS3, y las diferentes interacciones se programaron con la ayuda de JavaScript.

Para la interacción con el usuario se realizaron tres archivos HTML, el primero llamado signin.html que muestra una sección para que el usuario pueda ingresar su nombre de usuario y contraseña asignada, de manera predeterminada el usuario es “admin” y la contraseña “admin” la cual puede ser modificada una vez ingresado al sistema, esta primera página se realizó con el objetivo de validar al usuario que ingresa al sistema y brindarle seguridad al sistema.



Figura 2.12: Login de usuario.

En la Figura 2.12 se muestra la maquetación de la página signin.html que se ha diseñado.

La segunda página que se realizó es index.html que contiene las siguientes partes:

- Menú principal:
 - ✓ Nombre del sistema (Logo),
 - ✓ Secciones o módulos (Programar, Reportes, Usuario)
 - ✓ Mensaje de bienvenida al usuario junto con la foto del mismo, y el link para salir del sistema.



Figura 2.13 Menú principal.

La Figura 2.13 muestra el menú principal y las partes antes mencionadas.

- Secciones de módulos.
 - ✓ **Programación de procesos:** Donde se podrá ingresar los datos del lote de cacao a secar, los procesos que se requieren y los tiempos de dichos procesos.

Figura 2.14: Sección de programación.

En la Figura 2.14 se puede observar la sección de programación que se ha maquetado en el sistema.

- ✓ **Reportes:** Donde se visualizaran los reportes de las programaciones de secado anteriores, y se puede realizar búsquedas en la base de datos.

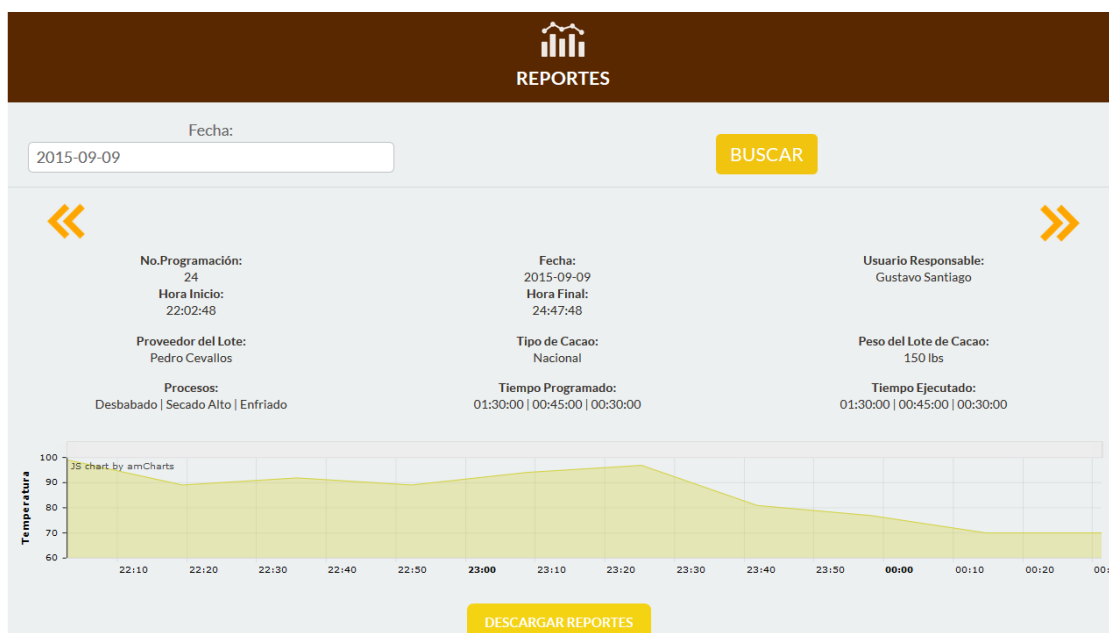


Figura 2.15: Sección de reportes.

En la Figura 2.15 se muestra la sección reportes del sistema.

- ✓ **Usuario:** Donde se puede visualizar los datos del usuario que acaba de ingresar al sistema, y según los roles de este usuario se puede visualizar la opción ingresar un nuevo usuario o editar sus propios datos.



Figura 2.16: Sección de Perfil de Usuario.

En la Figura 2,16 se muestra la sección que muestra el perfil del usuario.

La tercera página que se realizó, se llama executing.html la cual reemplaza la sección de programación de la página index.html por la sección de monitoreo:

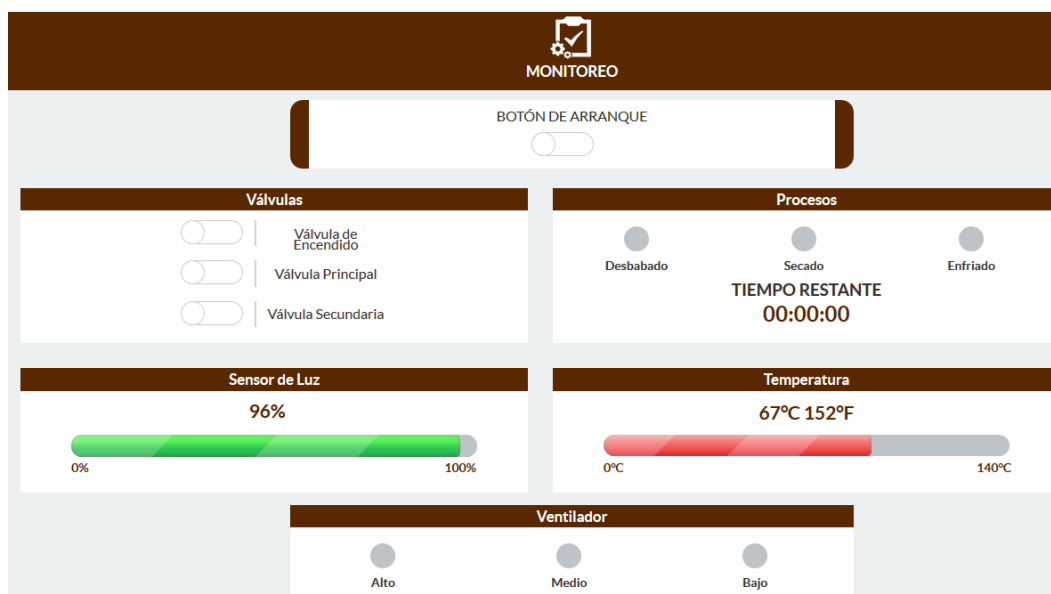


Figura 2.17: Sección de monitoreo.

En la Figura 2.17 se muestra la sección monitoreo de la página `executing.html`, la cual es una sección que contiene los controles para poder ejecutar las acciones programadas, y a la vez cuenta con los indicadores que muestran los procesos que están siendo ejecutados.

Las demás secciones como menú principal, reportes y usuario también se encuentran presentes en esta página tal cual se presentan en la página `index.html`.

2.5.3. Diseño del Controlador

Es el cerebro del sistema, el que controla los pines GPIO de salida y los sensores, debido a la facilidad que ofrece BoneScript y la compatibilidad con Node.js la programación del mismo estará escrita en JavaScript.

El controlador obedece en su mayoría a lo que está ocurriendo en la parte del cliente Web, las configuraciones que realiza, las válvulas que son activadas, los registros en la base de datos a los cuales quiere tener acceso, etc. por lo tanto necesitamos también una manera de comunicar el controlador con el cliente Web y permitir el intercambio de información entre ellos.

Puede analizarse el sistema en dos partes, el proceso de encendido automático con su respectivo método de apagado de emergencia en caso de no encender, y la parte que realiza el monitoreo y control de los procesos de secado programados por el usuario una vez que el encendido automático fue exitoso.

El usuario desde la interfaz puede programar un proceso de secado por tiempo o manejar directamente las válvulas a conveniencia (Modo Manual), sea cual fuere la decisión del usuario, el proceso de encendido automático siempre será el

mismo y se llevará a cabo una vez que el usuario active el botón de arranque.

El proceso de encendido conlleva **activar la válvula de encendido**, la cual es una válvula que expulsa gas con una presión menor a las demás válvulas, y **producir la chispa eléctrica** por medio de una bujía la cual en conjunto con la válvula de encendido deberá encender una pequeña llama piloto.

Es gracias a esta pequeña llama que podemos encender las demás salidas de gas dependiendo de lo programado por el usuario.

Para todo proceso realizado en nuestro sistema, se ha considerado un tiempo de **20 segundos** para permitir que la recámara de combustión encienda usando la pequeña llama del encendido automático, utilizando el sensor de luz podremos detectar si esto ha ocurrido, si la recámara no enciende, el sensor detecta alrededor de 1% de luminosidad, mientras que si la recámara logra encender detectará entre un 20% - 30% de luminosidad.

El sensor está colocado estratégicamente en un lugar donde no se concentra tanto calor como en la recámara (detrás de las válvulas) y la temperatura no excede los 40°. El **sistema de apagado de emergencia** consiste en cerrar todas las válvulas de gas y apagar el ventilador si en **20 segundos no ha encendido la recámara de combustión**, esto para evitar que el gas siga acumulándose en la recámara y ocurra algún accidente.

Si el sistema es programado por el usuario se ha definido la configuración de válvulas y velocidad del ventilador a utilizar para alcanzar una temperatura óptima para el proceso seleccionado, cabe destacar que el sistema no regula el paso del gas conforme incrementa o disminuye la temperatura sino que estos son

valores establecidos con los cuales trabajaba la secadora anterior para alcanzar un rango de temperaturas ideal para el proceso de secado del cacao.

Procesos de Secado Programables	PROCESO	VALVULA 1	VALVULA 2	VENTILADOR			TEMPERATURA APROXIMADA
				ALTO	MEDIO	BAJO	
	DESBABADO	●	●	●			100 °C
	SECADO ALTO	●			●		80 °C
	SECADO BAJO	●				●	70 °C
	ENFRIADO			●			40 °C

Tabla 2: Configuración del sistema para cada proceso.

La Tabla 2 muestra la configuración de válvulas y velocidad del ventilador que hemos definido para cada proceso y la temperatura aproximada deseada.

Una vez realizadas las pruebas, retroalimentaremos esa información para poder mejorar el sistema y conseguir un proceso óptimo de secado puesto que es necesario conocer el tiempo de funcionamiento, la temperatura, el peso del cacao y la humedad del cacao después de haber sido secado para saber si el proceso fue exitoso y en caso de no serlo ajustar el tiempo o la temperatura para que esto ocurra.

A través del módulo de reportes se puede conocer toda esta información y para conocer la humedad del cacao la debe comprobar una persona que confirme la cantidad de humedad **dentro** de la almendra y una vez realizado esto se puede determinar si se necesita más tiempo de secado o disminuir la temperatura si el cacao perdió demasiado peso.

El software controlador utiliza las configuraciones predefinidas en la Tabla 2 y realiza el debido proceso en el tiempo que ha determinado el usuario desde la interfaz web, monitoreando constantemente si la recámara sigue encendida y mostrando en pantalla el valor de temperatura (actualizado cada 5 segundos).

El controlador también responde correctamente en caso de que el usuario decida trabajar con la secadora de manera manual y no programada por tiempo, activando o desactivando correctamente las válvulas en el momento en que el usuario lo decida desde la interfaz web.

Cabe recalcar que el usuario **no tiene control sobre el proceso de encendido automático ni el sistema de apagado de emergencia**, ambos son transparentes para el usuario y no podrán ser modificados de ninguna manera.

Por último, el controlador desactiva todas las salidas activadas cuando se han realizado todas las tareas programadas y guarda un registro de todos los procesos ocurridos. Sumado a esto la interfaz web cuenta con un **botón de parada de emergencia que detiene todas las actividades** que se estén realizando en la secadora y guarda el registro de todos los procesos ocurridos hasta el momento y los que no pudieron ser realizados en la base de datos.

El servidor Web y la base de datos se pueden manejar desde la misma BeagleBone Black de tal manera que no necesitamos de ningún servidor externo para implementar el sistema.

No vamos a utilizar apache o nginx los cuales son software de código abierto que generalmente se utilizan para levantar un servidor Web en plataformas Linux sino que vamos a utilizar Node.js que es una plataforma de código abierto escrita en JavaScript la cual contiene librerías que permiten que el

programa desarrollado actúe como servidor Web permitiendo la implementación del controlador en el mismo código fuente que levanta al servidor Web.

Resulta que BoneScript, una de las librerías de BeagleBone utilizada para controlar los pines GPIO de la tarjeta, está basada en Node.js lo cual implica que nuestro **controlador usa librerías de Node.js**, es por esta razón que hemos decidido trabajar todos los módulos que incluyen al controlador y el servidor web con librerías basadas en Node.js sin necesidad de preocuparnos por la compatibilidad entre diferentes middlewares o librerías.

Específicamente utilizamos **Express**, un entorno de trabajo con todas las herramientas necesarias para trabajar con aplicaciones Web.

El sistema que hemos diseñado necesita una autenticación por parte del cliente en la página web y para esto utilizamos librerías de **Passport** enfocadas a la autenticación usando la base de datos de MySQL integrada en la BeagleBone que nos permitirá confirmar si las credenciales son válidas. El uso de este middleware nos permite encapsular la parte de autenticación de usuario, volviendo el sistema más robusto puesto que todo el proceso de autenticación es transparente para el cliente Web.

Para evitar problemas con la inyección de código SQL hemos decidido utilizar **Bookshelf**, que nos permite crear un Objeto en JavaScript a partir de las tablas creadas en la Base de Datos, **representándola virtualmente como un Objeto** y pudiendo así acceder a los datos guardados en la Base y manipularlos sin necesidad de conectarnos a la Base directamente y ejecutar consultas sobre ella.

Como mencionamos anteriormente necesitamos una manera de comunicar el cliente Web con el servidor y el controlador, y para

esto vamos a trabajar con **Socket.IO**, que nos permite tener una comunicación bidireccional basada en eventos.

Socket.IO básicamente nos otorga librerías para el envío y recepción de datos en la parte del Servidor Web y el Cliente Web, pero el Cliente Web también posee un script que establece el comportamiento de la página HTML cuando recibe datos a través de un socket y como enviar los datos al Servidor Web si ocurrió un evento determinado desde la interfaz web.

El servidor debe enviar información de registros guardados en la base de datos e información del usuario conectado hacia el Cliente Web, y el controlador envía constantemente información de los sensores, válvulas y procesos en los que se encuentra actualmente el cliente Web.

Esto permite que lo que ocurre en el controlador y el servidor sea visible desde la interfaz web de una manera amigable con el usuario y así mismo permite al usuario interactuar con el controlador y guardar datos en la base dese una interfaz sencilla de utilizar y no a través de una línea de comandos o funciones en JavaScript, facilitando grandemente el uso de la aplicación mediante botones y alertas sencillas de comprender e interactuar.

2.6. Implementación del software

El software fue implementado en el mismo controlador que maneja el hardware, es decir la BeagleBone Black, la interfaz de usuario del sistema puede ser vista desde cualquier explorador o navegador, conectándose al servidor web que se encuentra instalado en el controlador, utilizando la dirección IP de la BeagleBone Black o el nombre de dominio www.ssc.com puesto que la BeagleBone Black también cuenta con un servidor DNS que hemos configurado para

permitir al usuario un acceso más sencillo sin necesidad de recordar direcciones IP.

Todo lo anterior representa una conexión local por lo tanto cualquier dispositivo que necesite conectarse con el servidor Web deberá estar en la misma red local en la que se encuentra la BeagleBone Black.

Para conexiones remotas desde Internet la BeagleBone debe tener una salida a Internet y es posible conectarse a ella a través de la dirección IP Pública correspondiente, en este caso sería necesario activar redirección de puertos en el enrutador que se conecta a Internet (podría ser necesario contactar a su proveedor de servicios de Internet para activar redirección de puertos).

2.7. Diseño de la Red de Comunicación

El usuario final tiene la opción de conectarse al sistema de secado de manera local y remota. Para que nuestro sistema tenga esta característica debemos realizar un diseño de red que nos permita realizar esta conexión y que no sea difícil de configurar para el usuario final.

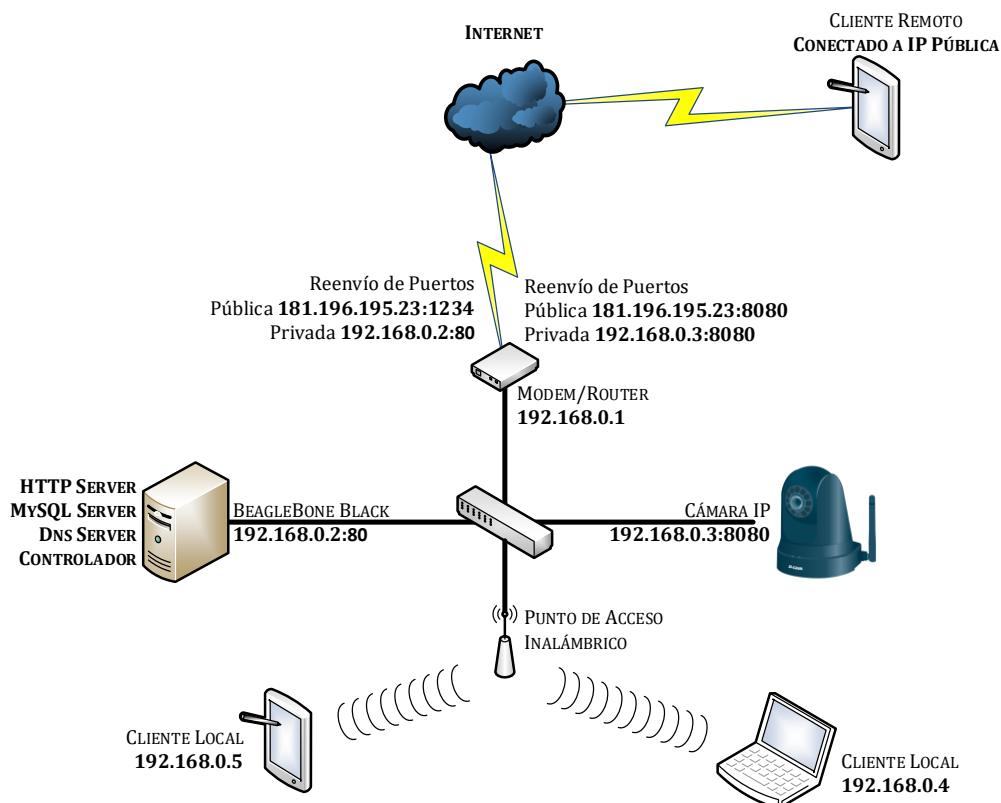


Figura 2.19: Esquema de red para conexión local y remota

- **Conexión local**

En la Figura 2.19 podemos observar el esquema de red propuesto, la BeagleBone Black, la Cámara IP y el Cliente Local están ubicados en el mismo segmento de red 192.168.0.0/24, los dispositivos utilizados para conectarlos pueden ser un conmutador y un Punto de Acceso Inalámbrico o puede tratarse también de un Enrutador Inalámbrico, lo importante es que estos dispositivos se encuentren en el mismo segmento de red.

Nuestro Modem/Enrutador será además el servidor DHCP y entregará direcciones IP para los diferentes Clientes que se conecten localmente, la BeagleBone Black y la Cámara IP

tendrán direcciones IP reservadas tal y como se indica en la Figura 2.19.

La BeagleBone Black tendrá un servidor DNS para que el usuario pueda acceder de manera local al sistema mediante la página www.ssc.com, sin necesidad de tener que recordar su dirección IP.

Cabe recalcar que la Cámara IP está disponible por medio del servidor web para que el usuario pueda acceder a ella aunque no recuerde la dirección IP de la misma.

- **Conexión Remota**

Para la conexión remota se va a utilizar la dirección IP pública entregada por el Proveedor de Servicios de Internet, configurando NAT para poder acceder al Servidor Web de la BeagleBone Black a través de 192.168.0.2:80, y de la misma manera a la cámara IP en la dirección 192.168.0.3:8080, de ésta manera el usuario puede conectarse desde Internet a través de la dirección IP Pública y el puerto correspondiente al dispositivo que intenta acceder.

- **Seguridad en la conexión**

Surge una duda al momento de revisar el diagrama de red de la Figura 2.19, ¿Por qué no utilizar una VPN para conectar al cliente remotamente?, sin una VPN los datos están siendo transmitidos en texto plano, fáciles de capturar y leer para un intruso en la red.

La respuesta es que, debido a que este proyecto está orientado al sector agrícola, quien utilice nuestro sistema tendrá poco o nada de conocimiento de redes, por tanto será una dificultad para el o los usuarios realizar una conexión al servidor VPN, y la idea es que el sistema pueda usarse desde cualquier dispositivo con

tan solo conectarse a la red, tampoco es factible dejar uno o dos dispositivos con la conexión a la VPN configurada, ya que si nuevos dispositivos tratan de conectarse con el sistema no tendrán la configuración apropiada y el sistema no sería escalable, por eso presentamos una solución que permita la escalabilidad de nuestra sistema en términos de permitir a nuevos usuarios utilizar sin problemas y de manera segura la interfaz web.

Debido a estas complicaciones al momento de conectarse es que no utilizamos una VPN, pero no podemos despreocuparnos del fallo de seguridad que esto representa.

Debemos impedir que los datos viajen de manera insegura a través de la red, tomando en cuenta que la información más importante que se envía a través de la red es el usuario y contraseña para acceder al sistema.

Y para evitar que esa información sea observada por terceros el Servidor Web **utilizará el protocolo HTTPS**, logrando así que la conexión esté protegida ante ataques “Man in the Middle”, ya que si alguien observara el tráfico en la red no podría entender la información transmitida puesto que sólo vería datos cifrados.

Para utilizar HTTPS en el Servidor Web que hemos creado basado en Express, debemos utilizar la librería HTTPS y especificar en qué puerto vamos a escuchar por conexiones entrantes, nosotros hemos utilizado el puerto predeterminado para HTTPS (443).

- **HTTP Strict Transport Security**

Ya hemos establecido nuestra conexión por medio del protocolo HTTPS pero nuestro servidor aún no es del todo seguro debido a que es posible establecer una comunicación “ignorando” el

protocolo HTTPS a través de ataques tales como HTTPS Stripping utilizando la herramienta “SSLStrip”, o Session Hijacking.

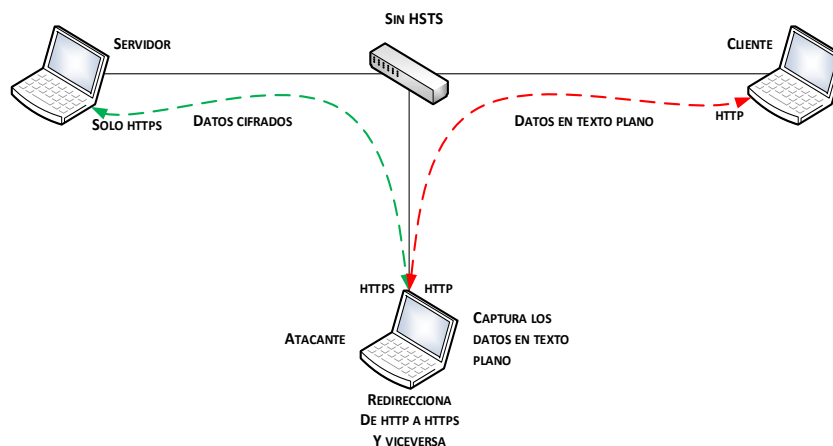


Figura 2.20 Ataque MITM sin HSTS

En la Figura 2.20 se evidencia claramente el tipo de ataque, el cliente se conecta al atacante usando HTTP y como consecuencia el atacante podrá leer los datos sin problemas, luego el atacante redirige los paquetes usando HTTPS para que la conexión con el Servidor Web no sea interrumpida.

Para poder mitigar estos posibles ataques hemos decidido implementar **HSTS (HTTP Strict Transport Security)**.

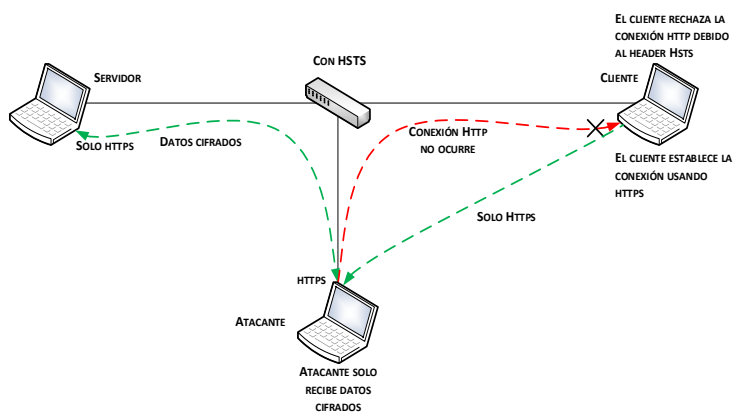


Figura 2.21 Ataque MITM con HSTS

HTTP Strict Transport Security es una política de seguridad que indica al Servidor Web que envíe una cabecera extra indicando al Cliente Web que solo puede establecer conexión utilizando HTTPS y especificando un período de tiempo durante el cual el cliente puede acceder al servidor de manera segura.

Tal y como se indica en la Figura 2.21, una vez que el navegador del Cliente Web reciba la cabecera automáticamente **rechazará cualquier tipo de comunicación vía HTTP** e inmediatamente establecerá la conexión a través de HTTPS.

Para implementar esta política de seguridad es necesario instalar y utilizar la librería Helmet para Node.js e indicar a nuestro Servidor Web que envíe dicha cabecera además de otros parámetros configurables.

En resumen, el cliente se conectará al Servidor a través de HTTPS enviando sus credenciales de manera cifrada ya sea en una red local o de manera remota, y en caso de ser víctima de un ataque Man in the Middle que trate de interceptar las credenciales de inicio de sesión no se logrará interceptar más que datos cifrados.

Un sistema nunca estará 100% libre de ataques informáticos pero creemos que debido a que los datos involucrados en este proyecto no son tan sensibles, es más que suficiente con proteger nuestro sistema ante estos ataques de interceptación de credenciales.

2.8. Costos de Implementación

A continuación presentaremos el análisis de costos del sistema de telecontrol y telemetría sin incluir los gastos que generaría implementar la secadora de cacao completa, ya que eso varía según el modelo y capacidad de este artefacto

DISPOSITIVO	COSTOS (\$)
BeagleBone Black	70,00
Sensor de Temperatura (RTD PT 100)	10,00
Sensor de Luz (LDR)	1,00
Tarjetas Electrónicas (Circuito impreso e implementos)	60,00
Panel Metálico	26,00
Router Inalámbrico (D-Link 5010)	20,00
Cámara Ip (D- Link L5010)	82,00
TOTAL	269,00

Tabla 3: Resumen de Costos de Implementación

En la tabla 3 podemos observar que el sistema completo tiene un costo de \$269, lo cual nos indica que este sistema es relativamente bajo en comparación a un sistema que utiliza como controlador un PLC, cuyo costo solo de este dispositivo es de \$300 a \$400, sin contar con demás dispositivos que deben integrarse para que llegue a tener las características del sistema que se ha propuesto.

Cabe mencionar que en este análisis se encuentra incluido el costo de la cámara de video, el cual es de \$82 el que puede ser obviado o amenorado, para que el sistema tenga los requerimientos mínimos para que la secadora realice los procesos sin afectar su desempeño, reduciendo así notablemente el costo total del sistema.

CAPÍTULO 3

3. ANALISIS DE PRUEBAS Y RESULTADOS

En el presente capítulo se presentan cuatro escenarios con el que se intenta simular procesas en una maqueta elaborada para verificar los resultados y comprobar el funcionamiento correcto de nuestra solución.



Figura 3.1: Elaboración de maqueta.

En la Figura 3.1 se muestra una imagen de la maqueta que se elaboró para realizar los respectivos escenarios que se presentan a continuación.

3.1. Escenario 1

En este primer escenario se va a realizar el proceso de secado utilizando el sistema de monitoreo y realizando la tarea específica de desbabado por 01h30min, secado alto por 00h45min y enfriado por 00h30min para luego ver los resultados y analizar si los módulos del sistema funcionan correctamente.

3.1.1 Resultados

Se muestra a continuación la Figura 3.2 donde se describen los procesos que se monitorearon y sus diferentes etapas.



Figura 3.2: Resultados del escenario 1.

- **Análisis del escenario 1**

En la Figura 3.2 se muestra el reporte de los procesos que se ejecutaron según lo programado, como podemos observar, el tiempo programado para cada proceso es igual al tiempo ejecutado de los mismos, también podemos observar que la curva de temperatura vs tiempo decae notablemente en el proceso de enfriado, lo cual ocurriría en un ambiente real.

3.2. Escenario 2

En este segundo escenario se va a realizar el proceso de secado utilizando el sistema de monitoreo y realizando la tarea específica de secado alto por 01h00min y enfriado por 00h20min para luego ver los resultados y analizar si los módulos del sistema funcionan correctamente.

3.2.1. Resultados

Se muestra a continuación las Figuras donde se describen los procesos que se monitorearon y sus diferentes etapas.



Figura 3.3: Resultados del escenario 2.

- **Análisis del escenario 2**

En la Figura 3.3 se muestra el reporte de los dos procesos que se ejecutaron sin ningún problema como se puede observar en el reporte, el tiempo programado para cada proceso es igual al tiempo ejecutado de los mismos, y en el gráfico de temperatura los valores se encuentran a los 80°C como correspondería a un proceso de secado alto en un ambiente real.

3.3. Escenario 3

Este tercer escenario se lo va a realizar para comprobar el sistema de seguridad contra accidentes, para lo cual se va a programar un proceso de desbabado por solo 20 segundos, secado alto por 20 segundos y enfriado por 10 segundos, y en el segundo 20 se simulará que se apagó la llama en la recámara de combustión para así ver en el reporte la diferencia entre el tiempo programado y el ejecutado.

3.3.1. Resultados

Se muestra a continuación las Figuras donde se describen los procesos que se monitorearon y sus diferentes etapas.



Figura 3.4: Resultados del escenario 3.

- **Análisis del escenario 3**

Como se puede observar en la Figura 3.4 el proceso de desbabado de 20 segundos se ejecutó sin anomalías, sin embargo a partir de los 20 segundos el sistema se detuvo por lo que muestra una gráfica de temperatura truncada ya que todo el sistema de monitoreo dio por finalizado todos los procesos lo cual se puede comprobar también comparando los tiempos de programación con los tiempos de ejecución, donde los procesos siguientes dicen que el tiempo de ejecución fue de cero.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. El prototipo permite el telecontrol y telemetría de los procesos ejecutados en una secadora de cacao mediante una interfaz web utilizando herramientas de software libre ahorrando así alrededor de un 50% en costos de implementación frente a un sistema de telecontrol similar que utiliza un PLC.
2. El proceso de secado del cacao mejora notablemente debido a la retroalimentación que brinda la sección de reportes que contiene un historial de temperatura vs tiempo de secado, y permite al usuario mejorar el secado de su producto continuamente basado en sus experiencias anteriores.
3. La accesibilidad que brinda el sistema permite al usuario mantenerse pendiente del estado de la secadora y mediante la cámara vigilar constantemente todo el sistema mientras se encuentra realizando otras actividades en el campo.
4. El sistema de alertas y seguridad contra accidentes responde de manera eficaz ante cualquier peligro en lo respecta al cierre de las electroválvulas que dan paso a la salida de gas.
5. La interfaz web es adaptable a cualquier dispositivo inteligente, intuitivo de operar, además de amigable con el usuario lo que da la facilidad para que pueda explorar todas las funciones integradas en el sistema, incluso para agricultores que no utilizan frecuentemente tecnología de punta.

Recomendaciones

1. En caso de necesitar una cobertura de red local amplia, se recomienda mejorar la antena del dispositivo de red inalámbrico que se esté utilizando.
2. Para conectarse remotamente desde Internet, es necesario que el proveedor de Servicios de Internet habilite reenvío de puertos para poder realizar la configuración adecuada.

3. Se recomienda apagar la BeagleBone Black una vez que los procesos de secado hayan terminado, puesto que el encendido constante desgasta la vida útil del dispositivo, como ocurre con cualquier otro dispositivo electrónico.
4. Se puede utilizar un adaptador de red inalámbrica en la BeagleBone Black para evitar el uso de un enrutador, pero hay que tener en consideración que el rango de cobertura del adaptador será mucho menor.
5. Al sistema se pueden añadir más sensores que permitirían tener un mejor control de las operaciones realizadas en la secadora, como es el caso de una balanza electrónica que pese el cacao una vez secado y lo guarde automáticamente en los reportes para conocer cuánto peso perdió el cacao al secarse.
6. Si la secadora tuviera un tamaño mayor para aumentar la capacidad de quintales de cacao que puede secar, se recomendaría utilizar varios sensores distribuidos dentro de la recámara para conocer el valor de temperatura promedio.

BIBLIOGRAFÍA

- [1] J. L. Maza. (2015, Septiembre 1). Secadero de Cacao [Online]. Disponible en: <http://www.ecocacao.ec>
- [2] D. E. Constante, "Diseño e implementación de un sistema de monitoreo remoto de una secadora de cacao de cilindro giratorio," Proyecto de Ingeniería, Facultad. Ing. Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito, Ecuador, 2012.
- [3] S. A. Guillen (2015, Septiembre 1). Procesos de Selección de Cacao [Online]. Disponible en: <https://www.youtube.com/watch?v=r7X8cY4QAYI>
- [4] T. Dicola (2015, Septiembre 5). Comparativa de tarjetas Linux embebidas [Online]. Disponible en: <https://learn.adafruit.com/embedded-linux-board-comparison/overview>
- [5] Texas Instruments Incorporated (2015, Septiembre 5). AM335x Sitara Datasheet [Online]. Disponible en: <http://www.ti.com/lit/ds/symlink/am3358.pdf>

ANEXOS

MANUAL DE USO DE BEAGLEBONE BLACK

CONEXIÓN Y PROGRAMACION DE LA BEAGLEBONE

La tarjeta BeagleBone viene con los drivers necesarios para su configuración, tan solo es necesario conectarla vía cable mini-USB e inmediatamente la computadora la reconocerá como un dispositivo de almacenamiento externo.

Entre las carpetas almacenadas en la memoria interna de la BeagleBone existe una llamada **drivers** que contiene los archivos necesarios para la instalación de un driver que básicamente permite a la BeagleBone usar el puerto USB para comunicación de datos Ethernet.

Se deberá escoger el driver correspondiente para entornos Windows, Linux, o Mac y una vez instalados correctamente, se creará un adaptador virtual en el computador que establecerá una conexión punto a punto con la BeagleBone a través del USB.

CONEXIÓN A LA BEAGLEBONE BLACK A TRAVES DE LA WEB

Solo se necesita abrir un navegador web y escribir la dirección IP de nuestra BeagleBone, se recomienda el uso de Mozilla Firefox o Google Chrome debido a que otros exploradores no poseen algunas funcionalidades en JavaScript necesarias para el servidor web de la BeagleBone.

En el servidor web de la BeagleBone se encuentra la documentación correspondiente, por ejemplo:

- Instrucciones sobre la instalación de otro sistema operativo a la memoria flash interna.
- Información acerca de los pines GPIO y como configurar sus diferentes funcionalidades.
- Documentación de la librería **BoneScript** que permite operar los pines GPIO con gran facilidad a través de sus funciones.

- Descripción de algunas funciones de JavaScript útiles para el manejo de GPIO.

UTILIZACION DE CLOUD9 COMO IDE PARA PROGRAMAR LA BEAGLEBONE

La tarjeta cuenta con un IDE (Integrated Development Enviroment) llamado **Cloud9**, incorporado en la interfaz web, que permite programar la BeagleBone con extrema facilidad, para acceder a él es necesario conectarse vía web al **puerto 3000** (192.168.7.2:3000).

El código escrito en Cloud9 es copiado a la BeagleBone a través de SSH y el IDE permite hacer depuración del código lo cual facilita la corrección de errores de sintaxis y errores lógicos.

También posee una terminal que es utilizada para controlar la BeagleBone Black directamente desde la web.

CONEXIÓN A LA BEAGLEBONE A TRAVES DE SSH

Para conectarse por SSH desde Windows se debe usar alguna aplicación para poder iniciar la sesión SSH, en nuestro caso usamos PuTTY, la cual es una aplicación gratis y de fácil uso.

Al abrirla se indica la dirección IP del dispositivo donde vamos a iniciar la sesión, el cual es 192.168.7.2.

```
login as: root
Debian GNU/Linux 7

BeagleBoard.org BeagleBone Debian Image 2014-04-23

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
Last login: Wed Apr 23 20:20:11 2014
root@beaglebone:~# █
```

Figura 1: Conexión a BeagleBone a través de SSH

Como se muestra en la Figura 1, el usuario es **root** y no hay contraseña (esto puede modificarse después). Después de ello, se presenta la consola de la misma manera que si abriéramos una terminal en Linux.

Si estamos usando Linux, la conexión con el dispositivo es muy simple, solo tenemos que abrir una terminal en Linux, y escribir el siguiente comando “`ssh 192.168.7.2 -l root`”.

De esa manera establecemos conexión con el dispositivo 192.168.7.2 y la bandera -l indica el usuario con el cual nos estamos conectando.

Como dato adicional, todo esto se realiza con la BeagleBone conectada por medio de USB, pero es posible conectarse mediante SSH con la BeagleBone conectada a través de Ethernet, en tal caso debemos conocer la dirección IP que se asignará a la tarjeta Ethernet de la BeagleBone puesto que será diferente a 192.168.7.2 pero el procedimiento es el mismo.

Una vez iniciada la sesión SSH tenemos la misma libertad de realizar acciones en la BeagleBone tal y como si estuviéramos configurándola directamente, es decir podemos crear archivos, carpetas, ejecutar scripts, etc. Aunque obviamente SSH nos presenta solo una línea de comandos, por lo tanto no podremos usar todo tipo de aplicaciones que utilicen interfaz gráfica como nuestro IDE desde SSH.

CONEXIÓN A LA BEAGLEBONE DE MANERA DIRECTA

Como habíamos mencionado la BeagleBone es una micro computadora y por lo tanto podemos conectarnos directamente a ella sin necesidad de iniciar sesiones SSH o vía Web. Para ello necesitamos lo mismo que cualquier otra PC necesita, un monitor, un teclado y un mouse.

Debido a que la BeagleBone solo trae un puerto USB se necesita colocar un USB Hub que nos permitirá conectar varios dispositivos USB a ella como son el teclado y el mouse.

Para conectar el monitor la BeagleBone trae un puerto Micro HDMI y puesto que nuestro monitor posiblemente sea VGA, HDMI o DVI usamos el respectivo adaptador para poder conectarlos.

Una vez tenemos todo listo, usando nuestro adaptador AC conectamos la BeagleBone y automáticamente iniciará Debian, nuestra distribución de Linux incorporada en la BeagleBone de fábrica.

Cloud9 aún puede ser utilizado abriendo el explorador web y escribiendo la dirección **127.0.0.1:3000** o **localhost: 3000**.

INSTALACION DE UN NUEVO SOFTWARE EN LA BEAGLEBONE

Existen diferentes maneras de instalar software no tanto en la tarjeta BeagleBone sino más bien en entornos Linux, y puesto que nuestra BeagleBone tiene instalado Debian todo nuevo software viene en “paquetes” que pueden ser instalados mediante un solo comando.

Para descargar e instalar estos paquetes se necesita un gestor de paquetes, el cual se encarga de instalar no solo el software que deseamos sino también las **dependencias**, que no son otra cosa que software necesario para que la aplicación que deseamos instalar funcione correctamente.

En distribuciones basadas en Debian el gestor de paquetes es **apt** y antes que nada, es recomendable actualizar la lista de paquetes disponibles que el gestor posee, esto se hace mediante el siguiente comando “**sudo apt-get update**”.

La tarjeta debe estar conectada a Internet para que este procedimiento pueda tomar lugar, basta con conectar mediante el puerto Ethernet a nuestro Router para que esto suceda, ya que por lo general el Router que tenemos en casa de nuestro proveedor de Internet reparte direcciones IP de manera automática mediante protocolo DHCP.

Una vez haya terminado de actualizarse la lista de paquetes podemos instalar software mediante el siguiente comando “**sudo apt-get install <nombre_de_la_aplicación>**”

APAGADO CORRECTO DE LA BEAGLEBONE

Por medio de SSH también podemos apagar e incluso reiniciar la BeagleBone utilizando la línea de comandos, cabe destacar que desconectarla de la energía

podría provocarle daños y pérdida de información ya que sería el equivalente a desconectar una PC de la fuente de energía.

Para apagarla utilizamos el siguiente comando `sudo shutdown -h now`.

Y para reiniciarla `sudo shutdown -r now`.

La diferencia entre apagarla y reiniciarla reside en el parámetro que le estamos enviando al comando shutdown, HALT corresponde a `-h` y REBOOT a `-r`.

El parámetro "now" es un parámetro de tiempo, técnicamente podríamos indicar al sistema un tiempo específico en el cual deseamos que se apague o se reinicie pero como en este caso queremos que se ejecute inmediatamente escribimos "now".

Como consejo podemos indicar que la línea de comandos posee una función de autocompletado de comandos, utilizando la tecla Tab, la consola autocompletará el comando o nos mostrará una lista con los posibles comandos, y también sirve para autocompletar directorios lo cual es muy útil cuando el nombre de éstos es bastante largo.

CONTROL DE LOS PINES GPIO DE LA BEAGLEBONE BLACK

Existen diversas formas de controlar los pines GPIO de la tarjeta utilizando librerías en Python, JavaScript, C++ etc., pero lo que hacen estas librerías es simplificar el procedimiento que detallaremos a continuación, la modificación de los directorios GPIO dentro del sistema operativo, más adelante veremos cómo se utilizan las librerías y cómo nos facilitan el procedimiento al momento de manipular los GPIO.

En primer lugar necesitamos acceder como usuario root y para eso utilizaremos el siguiente comando `sudo su`.

Después nos ubicaremos en el directorio GPIO de la siguiente manera `cd /sys/class/gpio`

Si listamos el contenido del directorio nos aparecerá una ventana similar a la Figura 2.

```

root@beaglebone:/sys/class/gpio# ls
export  gpiochip0  gpiochip32  gpiochip64  gpiochip96  unexport
root@beaglebone:/sys/class/gpio# █

```

Figura 2: Directorio GPIO

Existen dos archivos que nos permiten manipular los GPIO los cuales son “export” y “unexport”, básicamente para poder trabajar con un GPIO primero hay que exportarlo y cuando ya no lo necesitamos se considera una buena práctica liberar el pin por medio del archivo **unexport** para que luego pueda ser utilizado con un diferente propósito.

Para exportar un pin es necesario escribir en el archivo export el número del GPIO, basándonos en el esquema de Headers, por ejemplo si queremos acceder al GPIO 67 que se encuentra en el pin 8 del Header 8 debemos exportarlo de la siguiente manera “**echo 67 > export**”.

El comando funciona sólo si nuestro directorio actual es /sys/class/gpio, e inmediatamente se creara un directorio llamado gpio67 el cual contiene algunos archivos que nos permitirán modificar el estado del pin como se observa en la Figura 3.

```

root@beaglebone:/sys/class/gpio# echo 67 > export
root@beaglebone:/sys/class/gpio# ls
export  gpio67  gpiochip0  gpiochip32  gpiochip64  gpiochip96  unexport
root@beaglebone:/sys/class/gpio# cd gpio67
root@beaglebone:/sys/class/gpio/gpio67# ls
active low  direction  edge  power  subsystem  uevent  value

```

Figura 3: Directorio de un GPIO exportado

De estos archivos destacan unos cuantos y vamos a explicar brevemente su función:

Active_Low.- Es un archivo que indica si el pin obedece lógica positiva o negativa.

- En lógica digital, la lógica positiva indica que un nivel electrónico ALTO equivale a un 1 digital y un nivel electrónico BAJO equivale a un 0 digital.

- La lógica negativa es todo lo contrario, un nivel electrónico ALTO equivale a un 0 digital, de ahí el nombre de “Active_Low” indicando que un nivel electrónico BAJO o “LOW” equivale a un valor digital Activo (1 digital).

Si queremos trabajar con lógica negativa, basta con escribir un 1 en el archivo `active_low`, pero eso solo activará lógica negativa en ese pin específico.

Direction.- Como su nombre indica, este archivo define la dirección del pin GPIO la cual puede ser `out` (salida) o `in` (entrada). Como ejemplo si queremos definir el GPIO 67 como salida usaremos el comando siguiente `“echo out > direction”`

También nos permite conocer el estado del pin en cualquier momento usando el comando `“cat direction”`

Como dato técnico quiero añadir que cuando exportamos un pin GPIO, el modo estándar en el que es exportado es como **ENTRADA**, esto es así para no provocar una salida indeseada a un pin que pueda activar por error un circuito externo.

Edge.- Este archivo nos provee de una función muy interesante y muy utilizada en circuitos digitales, la cual nos permite realizar acciones cuando ha ocurrido un “cambio de estado”, en la Figura 4 se muestra gráficamente.

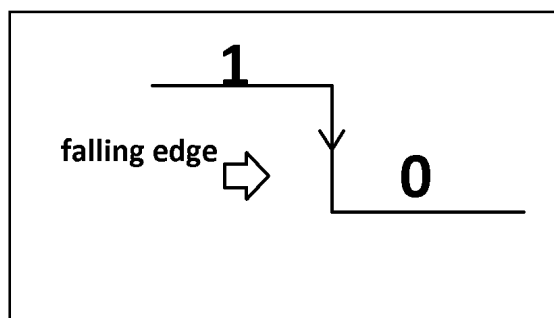


Figura 4: Explicación gráfica del cambio de estado lógico

Como lo muestra la Figura 4.4 un “Falling Edge” ocurre cuando se produce un cambio de estado de ALTO a BAJO, por otro lado un “Rising Edge” ocurre cuando el cambio es de BAJO a ALTO, para poder configurar estas acciones en el gpio se utilizan los siguientes comandos: `“echo rising | falling | change > edge”`

Lamará a una función cuando se produzca el cambio de BAJO a ALTO (rising), de ALTO a BAJO (falling) y un pulso completo (change).

De manera predeterminada un pin no verificará cambios de estado a menos que sea modificado el archivo edge.

Value.- Este archivo tiene dos versiones:

- Cuando el pin está configurado como ENTRADA, value tiene permisos de sólo lectura, de tal manera que solo podemos leer el valor de ENTRADA.
- Cuando el pin está configurado como SALIDA, value tiene permisos de lectura y escritura, para poder cambiar el valor de salida del pin y también conocer su valor actual.

Como cualquier archivo de texto, para leer su valor actual se utiliza el comando cat, y para escribir un valor se utiliza el comando echo.

CONTROL DE PINES GPIO USANDO LIBRERIAS

De manera breve vamos a explicar cómo es posible controlar los GPIO de una manera más sencilla e intuitiva utilizando librerías que ya vienen incluidas en nuestra BeagleBone las cuales son BoneScript que trabaja con JavaScript y Adafruit_BBIO utilizada en Python. También haremos una comparación entre ambas destacando la razón por la cual nos decidimos a trabajar con una de ellas.

BONESCRIPT

Librería diseñada para trabajar con los puertos GPIO de la BeagleBone basada en Node.js que es un entorno de trabajo escrito en JavaScript que nos permitirá usar una aplicación de JavaScript para interactuar a través de una interfaz web directamente con el servidor y permitir el control de los GPIO lo cual le da una gran ventaja en comparación con la librería de Python.

Para crear un script en JavaScript debemos crear un archivo de texto y añadirle la extensión “.js”, una vez hecho esto para cargar la librería BoneScript escribiremos lo siguiente: `var bone = require ('bonescript');`

En esta línea estamos creando un objeto en JavaScript llamado “bone” y estamos cargando los módulos de BoneScript a “bone”, en JavaScript no hace falta definir si una variable es entero, string o un objeto, el lenguaje automáticamente decide el tipo de variable que será.

CONFIGURACION DE UN PIN COMO ENTRADA O COMO SALIDA

Para poder indicar si un GPIO es de entrada o salida vamos a utilizar una función propia del objeto que adquirió los módulos de BoneScript, la función **pinMode ()**.

La función `pinMode ()` puede recibir hasta 6 parámetros pero por lo general usaremos los dos siguientes:

- Una cadena de caracteres con el nombre del pin que vamos a usar **referenciado al Header**, por ejemplo si queremos activar nuevamente el GPIO 67, observando la Figura 2.4, el GPIO 67 se encuentra en el Header P8 pin 8, entonces la forma correcta de definir el pin sería ‘P8_8’, debemos trabajar en conjunto con el esquema de Headers para saber si podemos utilizar ese GPIO y con qué nombre definirlo en el código fuente.
- El modo, el cual se define usando el objeto que hemos creado, de la siguiente manera

bone.OUTPUT Para modo salida

bone.INPUT Para modo entrada

Resumiendo, si deseamos utilizar el pin 8 del Header P8 como salida debemos hacerlo de la siguiente manera:

bone.pinMode ('P8_8', bone.OUTPUT);

ASIGNACION DE UN VALOR A PIN CONFIGURADO COMO SALIDA

Para modificar el valor de un pin GPIO configurado como salida tenemos la función **digitalWrite ()**.

Esta función recibe tres parámetros:

- Cadena de caracteres con el nombre del pin que vamos a modificar, ej. 'P8_8'.
- El valor que va a tomar el pin, que es una variable propia del objeto que creamos llamado "bone", puede ser bone.HIGH, o bone.LOW.
- Un callback, que en programación es un llamado a una función cuando haya terminado de ejecutarse digitalWrite (), este parámetro es opcional pero muy útil al momento de programar.

Podemos seguir con el ejemplo anterior y activar el pin P8_8 de la siguiente manera: **digitalWrite ('P8_8', bone.HIGH);**

Esta función además de cambiar el estado del pin, retorna un valor booleano True o False dependiendo de si la operación se completó exitosamente o no, ya que puede darse el caso que el pin que estamos tratando de activar esté configurado como INPUT y en ese caso solo es posible recibir información.

REVISION DEL VALOR DE UN PIN GPIO

Esto se puede realizar tanto para un pin de entrada como de salida, y la función a utilizar se llama **digitalRead ()**.

Esta función recibe dos parámetros:

- El identificador del pin que deseamos revisar, ej. 'P8_8'.
- Una función callback, que en este caso retornara el valor del pin (1 - 0).

CREAR UN INTERRUPTOR QUE SE EJECUTE CUANDO OCURRA UN EVENTO.

Esta es la gran ventaja que posee BoneScript y que marca la diferencia al momento de manejar eventos. Vamos a suponer que deseamos activar un LED si presionamos un botón, entonces debemos considerar lo siguiente.

- ✓ El LED estará conectado a un pin de SALIDA y apagado hasta que se presione el botón.
- ✓ El botón estará conectado a un pin de ENTRADA y recibiremos un 0 a menos que el botón sea presionado.
- ✓ Cuando el botón sea **presionado y soltado** el LED debe encenderse, en pocas palabras debemos cambiar el estado de esa salida a HIGH.
La función que nos permite crear este interruptor se llama **attachInterrupt ()** y recibe varios parámetros que explicaremos a continuación:

1. El identificador del pin que **provocará la interrupción**, ej. 'P8_8'.
2. Un booleano que indica si se ejecuta o no la interrupción, por lo general queremos que se ejecute así que colocamos "true", caso contrario podría colocarse una condición y la interrupción se provocará solo en caso de cumplirse la condición.
3. El modo del "**edge**" que explicamos anteriormente, puede ser RISING, FALLING o CHANGE, la opción aquí es CHANGE dado que el botón debe **presionarse y soltarse** para que el LED se encienda, en caso de poner RISING, se ejecutará en el momento que presionamos el botón, y si ponemos FALLING se ejecutará al soltar el botón.

El modo es una variable intrínseca de BoneScript por lo tanto lo definimos así:
bone.CHANGE

4. El último parámetro es la función a ejecutarse de producirse la interrupción, que para nuestro ejemplo sería una función que active el pin del LED.

Todo esto nos permite manejar los pines digitales de la BeagleBone sin problemas, en cambio las entradas analógicas usan funciones similares pero con diferente nombre como veremos a continuación.

MANEJO DE ENTRADAS ANALOGICAS

La BeagleBone Black tiene 7 convertidores Analógico-Digital (ADC), cuyo objetivo es tomar una señal analógica en el rango de 0 a 1.8V y escalarla a valores digitales discretos que van de 0 a 1.

Para leer el valor de una entrada analógica utilizamos la función **analogRead ()** la cual es muy parecida a `digitalRead ()`, y recibe dos parámetros:

- El identificador del GPIO, recordar que debe ser un pin analógico, por ejemplo el "P8_40".
- Una función callback que retorna el valor del pin en la variable "x.value".

También existe una función llamada **analogWrite ()** pero no debemos confundirla y usarla en un pin analógico debido a que estos pines solo pueden ser ENTRADAS.

La función `analogWrite ()` se usa en los pines **DIGITALES PWM** y de hecho no produce una salida analógica sino que simula una salida analógica mediante el uso de PWM (Pulse-Width Modulation) que es una técnica que con el uso de un filtro paso-bajo puede convertir una señal digital en una señal analógica.

Consta de 4 argumentos o parámetros los cuales son:

- El identificador del pin GPIO PWM, solo funciona con pines PWM.
- El ciclo de trabajo o duty cycle del PWM en valores de 0 a 1, el valor corresponde a un porcentaje, ej. 0.5 corresponde a un ciclo de trabajo de 50%.
- La frecuencia de la modulación en Hercios.
- Una función callback que se ejecutará al completarse la tarea de `analogWrite ()`.

Como podemos ver no definimos ningún valor de ALTO o BAJO debido a que el pin estará oscilando entre estado ALTO y BAJO a la frecuencia definida.

Hay muchos aspectos a considerar en cuanto a PWM se refiere pero resumiendo es una técnica para simular un valor analógico, recordemos que un pin digital puede estar en estado HIGH (3.3V) o LOW (0V), pero PWM nos permite simular un valor “analógico” intermedio. El valor intermedio dependerá del ciclo de trabajo y de la frecuencia a la que oscila la onda modulada.