



**ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ
MANUEL FÉLIX LÓPEZ**

CARRERA DE COMPUTACIÓN

**TESIS PREVIA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN INFORMÁTICA**

TEMA:

**DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE RIEGO
AUTOMATIZADO Y CONTROLADO POR UNA PLACA
ARDUINO PARA LA FINCA “LA LUCIA”**

AUTOR:

WILLIAM R. CERVANTES RODRÍGUEZ

TUTOR:

ING. FERNANDO RODRIGO MOREIRA MOREIRA, MBA.

CALCETA, JULIO 2016

DERECHOS DE AUTORÍA

William Rolando Cervantes Rodríguez, declara bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional, y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración se cede los derechos de propiedad intelectual a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, según lo establecido por la ley de Propiedad intelectual y su reglamento.

.....
WILLIAM R. CERVANTES RODRÍGUEZ

CERTIFICACIÓN DE TUTOR

Fernando Rodrigo Moreira Moreira certifica haber tutelado la tesis **DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE RIEGO AUTOMATIZADO Y CONTROLADO POR UNA PLACA ARDUINO PARA LA FINCA “LA LUCIA”**, que ha sido desarrollada por William Rolando Cervantes Rodríguez, previa la obtención del título de ingeniero en Informática, de acuerdo al **REGLAMENTO PARA LA ELABORACIÓN DE TESIS DE GRADO DE TERCER NIVEL**, de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

.....
ING. FERNANDO R. MOREIRA MOREIRA, MG.

APROBACIÓN DEL TRIBUNAL

Los suscritos integrantes del tribunal correspondiente, declaran que han APROBADO la tesis **DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE RIEGO AUTOMATIZADO Y CONTROLADO POR UNA PLACA ARDUINO PARA LA FINCA “LA LUCIA”**, que ha sido propuesta y desarrollada y sustentada por William Rolando Cervantes Rodríguez, previa la obtención del título de ingeniero en Informática de acuerdo al **REGLAMENTO PARA LA ELABORACIÓN DE TESIS DE GRADO DE TERCER NIVEL** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

.....
ING. ORLANDO AYALA PULLAS, Mg.
MIEMBRO DEL TRIBUNAL

.....
ING. MARLON NAVIA MENDOZA, Mg.
MIEMBRO DEL TRIBUNAL

.....
ING. DANIEL A. MERA MARTÍNEZ, Mg.
PRESIDENTE DEL TRIBUNAL

AGRADECIMIENTO

A la Escuela Superior Politécnica de Manabí Manuel Félix López que me dio la oportunidad de una educación superior y en la cual he forjado mis conocimientos profesionales día a día;

A los propietarios de la Finca La Lucia por las facilidades durante la implementación del sistema en sus instalaciones;

A los catedráticos por impartir sus conocimientos en clases durante el transcurso de la carrera como profesional, y

A mis Padres, a mi Esposa y mi mayor fuerza e inspiración mi Hija les agradezco por ayudarme a conseguir el sueño de ser un profesional.

.....
WILLIAM R. CERVANTES RODRÍGUEZ

DEDICATORIA

Quiero dedicar este trabajo a muchas personas que han sido muy importantes en mi vida entre ellos.

A Dios por darme la alegría de vivir y ser una persona para bien.

A mi hija Scarleth Lucia por ser la fuente de inspiración para seguir adelante con mis estudios.

A mis padres William Cervantes y Lucia Rodríguez por apoyarme en todo momento.

A mis hermanas y hermano por estar siempre apoyándome en todo momento y a mis sobrinos por la alegría que brindan en los momentos difíciles.

Dedicada a un compañero que aunque poco lo conocí siempre se destacó por luchar en sus creencias y sus valores haciendo respetar su posición de estudiante Rodolfo.

.....
WILLIAM R. CERVANTES RODRÍGUEZ

CONTENIDO GENERAL

| | |
|---|-----|
| DERECHOS DE AUTORÍA | ii |
| CERTIFICACIÓN DE TUTOR | iii |
| APROBACIÓN DEL TRIBUNAL..... | iv |
| AGRADECIMIENTO..... | v |
| DEDICATORIA..... | vi |
| CONTENIDO GENERAL..... | vii |
| CONTENIDO DE CUADROS | ix |
| CONTENIDO DE GRÁFICOS | ix |
| CONTENIDO DE FOTOS..... | x |
| RESUMEN | xi |
| PALABRAS CLAVE..... | xi |
| ABSTRAC | xii |
| KEYWORDS | xii |
| CAPÍTULO I. ANTECEDENTES | 1 |
| 1.1.PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA..... | 1 |
| 1.2.JUSTIFICACIÓN | 3 |
| 1.3.OBJETIVOS | 5 |
| 1.3.1. OBJETIVO GENERAL..... | 5 |
| 1.3.2. OBJETIVOS ESPECÍFICOS | 5 |
| 1.4.IDEA A DEFENDER..... | 6 |
| CAPÍTULO II. MARCO TEÓRICO..... | 7 |
| 2.1 CARACTERIZACIÓN DE LA FINCA “LA LUCIA” | 7 |
| 2.2. SISTEMA DE RIEGO | 7 |
| 2.2.1. RIEGO POR ASPERSIÓN | 8 |
| 2.2.2. INSUMOS DE IRRIGACIÓN | 9 |
| 2.3. TECNIFICACIÓN DE PROCESOS DE RIEGO..... | 9 |
| 2.4. DOMÓTICA..... | 10 |
| 2.4.1. ELECTRÓNICA..... | 10 |
| 2.4.2. AUTOMATIZACIÓN | 11 |
| 2.5. PLATAFORMA ARDUINO..... | 12 |
| 2.5.1. SOFTWARE | 12 |
| 2.5.2. HARDWARE | 13 |

| | |
|---|----|
| 2.5.3.4. DESCRIPCIÓN DE LA PLACA ARDUINO MEGA 2560..... | 17 |
| 2.5.3.5. SEÑALES DIGITALES Y SEÑALES ANALÓGICAS | 18 |
| 2.6. COMPONENTES A TRABAJAR CON PLACA ARDUINO | 19 |
| 2.6.1. ARDUINO PANTALLA LCD | 19 |
| 2.6.2. RTC ds1307 ARDUINO..... | 20 |
| 2.6.3. RELÉ PARA ARDUINO..... | 21 |
| 2.6.4. TARJETAS SD A 3.3v..... | 22 |
| 2.6.5. MICRO CONTROLADORES..... | 23 |
| 2.6.6. ELECTROVÁLVULA | 24 |
| 2.6.7. MICROPROCESADOR..... | 24 |
| 2.7. CIRCUITOS IMPRESOS..... | 25 |
| 2.8. MÉTODOS Y TÉCNICAS..... | 26 |
| 2.8.1. TÉCNICA DE LA ENCUESTA..... | 26 |
| 2.8.2. METODOLOGÍA HARDWARE LIBRE..... | 26 |
| CAPÍTULO III. DESARROLLO METODOLÓGICO | 33 |
| 3.1. MÉTODOS Y TÉCNICAS..... | 33 |
| 3.1.1. MÉTODO DEDUCTIVO..... | 33 |
| 3.1.2. TÉCNICA DE LA ENCUESTA..... | 33 |
| 3.1.3. TECNIFICACIÓN | 34 |
| 3.2. MÉTODO INFORMÁTICO..... | 34 |
| 3.2.1. MÉTODO HARDWARE LIBRE | 34 |
| 3.2.2. PROCESOS DE CONCEPTUALIZACIÓN | 34 |
| 3.2.3. PROCESO DE ADMINISTRACIÓN | 36 |
| 3.2.4. PROCESO DESARROLLO | 39 |
| CAPÍTULO IV. RESULTADOS Y DISCUSIÓN..... | 46 |
| CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES | 57 |
| 5.1 CONCLUSIONES..... | 57 |
| 5.2 RECOMENDACIONES | 58 |
| BIBLIOGRAFÍA | 59 |
| ANEXOS | 62 |

CONTENIDO DE CUADROS

| | |
|--|----|
| Cuadro 3.1 Análisis y reflexión sobre problemas y soluciones..... | 35 |
| Cuadro 3. 2 Definición del alcance de la investigación | 35 |
| Cuadro 3. 3 . Elaboración de la propuesta de desarrollo del proyecto | 35 |
| Cuadro 3. 4 Descripción del dispositivo a desarrollar..... | 36 |
| Cuadro 3. 5 Elaboración o actualización del plan del proyecto. | 37 |
| Cuadro 3. 6 Administración técnica del proyecto | 37 |
| Cuadro 3. 7 Especificación de hardware | 39 |
| Cuadro 3. 8 Programación de dispositivos | 41 |
| Cuadro 3. 9 Desarrollo de circuitos integrados..... | 42 |
| Cuadro 3. 10 Integración..... | 43 |
| Cuadro 3. 11 Integración del software en el hardware | 43 |
| Cuadro 3. 12 Verificación y Simulación..... | 44 |
| Cuadro 3. 13 Liberación del hardware y software | 45 |

CONTENIDO DE GRÁFICOS

| | |
|---|----|
| Figura 2. 1 Características de la placa Arduino..... | 15 |
| Figura 2. 2 Placa Arduino Chip de comunicación..... | 17 |
| Figura 2. 3 Placa Arduino Mega 2560 | 18 |
| Figura 2. 4 Pantalla Lcd para Arduino 16x2 Fuente Arduino (2013)..... | 19 |
| Figura 2. 5 Rtc ds1307 modulo para Arduino | 20 |
| Figura 2. 6 Relé 2 canales 5v Arduino | 21 |
| Figura 2. 7 Modulo lector de tarjeta Sd Arduino | 22 |
| Figura 2. 8 Micros controladores | 23 |
| Figura 2. 9 Válvula eléctrica | 24 |
| Figura 2. 10 Circuito Impreso | 25 |
| Figura 2. 11 El Proceso de desarrollo Fuente: Medrano (2011)..... | 27 |
| Figura 2. 12 Proceso detallado de conceptualización Fuente: Medrano (2011)..... | 28 |
| Figura 2. 13 Proceso de administración procesos de desarrollo de hardware libre | 29 |
| Figura 2. 14 Proceso de Desarrollo de Proyectos en Hardware Libre..... | 29 |

| | |
|---|----|
| Figura 2. 15 Especificación de Hardware Estático | 31 |
| Figura 2. 16 Programación de dispositivos | 31 |
| Figura 2. 17 Desarrollo de circuitos | 32 |
| | |
| Tabla 2. 1 Resumen descripción placa Arduino | 18 |
| | |
| Figura 3. 2 Diseño del control de riego y sus actuadores..... | 38 |
| | |
| Figura 4. 1 Tecnificación del controlador de sistema de riego..... | 46 |
| Figura 4. 2 Casos de uso (usuario del sistema) | 47 |
| Figura 4. 3 Diagrama de flujo de los procesos de riego | 48 |
| Figura 4. 4 Diagrama de flujo del proceso del control de riego | 49 |
| Figura 4. 5 Esquema de montaje electrónico | 50 |
| Figura 4. 6 esquema electrónico de los componentes que conforman el control de riego | 50 |
| Figura 4. 7 Diseño del circuito impreso | 51 |
| Figura 4. 8 Comprobación de la funcionalidad del código | 52 |
| Figura 4. 9 Simulación de reporte de procesos efectuados del control de riego | 54 |
| Figura 4. 10 Índice de horas del gasto de agua en el riego..... | 55 |

CONTENIDO DE FOTOS

| | |
|---|----|
| Foto 2. 1 tecnificación de procesos de riego | 10 |
| | |
| Foto 3. 1 Simulación efectuada en el control de riego..... | 44 |
| Foto 3. 2 Foto de la implementación del sitio | 45 |
| | |
| Foto 4. 1 Integración del código fuente al control de riego | 51 |
| Foto 4. 2 Baquelita circuito impreso | 52 |
| Foto 4. 3 Integración de los módulos a placa impresa | 53 |
| Foto 4. 4 Hardware instalado y probado | 56 |

RESUMEN

El objetivo de este trabajo fue crear un control automatizado de sistema de riego basado en una placa Arduino mega en la Finca “La Lucia” del sitio La Pita – Parroquia Canuto – Cantón Chone, lugar donde se permitió identificar la problemática y proponer la posible solución. El control automatizado del sistema de riego cuenta con varios dispositivos de hardware que complementado dicho control con teclado 4x3, Lcd de 2x12 donde el usuario accede al manejo del dispositivo de riego, mejorando el control en las actividades y procesos realizadas a diario en el área de los sembríos. Para la ejecución del control se trabajó con el método informático Hardware Libre que se usó para la elaboración de los procesos de conceptualización, administración y desarrollo en la implementación del control de riego, donde fue necesario recopilar información, determinar los sectores estratégicos para la instalación de los actuadores del riego, haciendo pruebas del control de riego en prototipos obteniendo como resultado el correcto funcionamiento del control. Finalmente se sometió a prueba en campo real donde el usuario verificó, y el autor demostró su buen funcionamiento, obteniendo los resultados favorables en el buen control del uso de agua en los sembríos de la Finca “La Lucia”.

PALABRAS CLAVE

Módulos Rtc, Modulo Relé, Arduino Mega, Sistema de riego, Tecnificación.

ABSTRAC

The author's aim was to create an automated irrigation control system based on an Arduino Mega, In "La Lucia" farm La Pita site - Canuto Parish - Chone Canton, place that allowed to identify the problem and to propose a possible solution. The automated irrigation control system has several hardware devices complemented with such control keyboard 4x3, 2x12 LCD where the user has access to the management of the irrigation device, improving the control of daily activities and processes carried out in the crops area. The planned execution of the control was worked on the computer method Free Hardware that was used for the development of the conceptualization processes, management and development in the implementation of irrigation control, where it was necessary to gather information, identify strategic sectors for the installation of irrigation controllers, doing tests of irrigation control in prototypes having as a result the right operation of the control. Finally a trial basis was set on the real field where the user verified, and the author showed its correct function, obtaining favorable results in the good control of water use in the crops at "La Lucia" farm.

KEYWORDS

Rtc modules, Relay Module, Arduino Mega, Irrigation system, Technification.

CAPÍTULO I. ANTECEDENTES

1.1. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA

El agua es tal vez el más importante de los recursos naturales, pues de él dependen la subsistencia de la población, la flora y la fauna, y la continuidad de la industria. En el actual estado del planeta, y con las presentes condiciones climáticas a escala global, este recurso y su gestión es causa de conflictos a escala local, regional, nacional e internacional.

Sin embargo, hay situaciones en las que no se le da un buen control o manejo de este líquido vital. Como en el caso de la agricultura, es entonces que los agricultores se ven en la necesidad de mejorar estos problemas más aun cuando esta situación se da en lugares donde el agua es escasa, como lo es en la zona de Canuto, La Pita, específicamente en la finca La Lucía. Siendo el principal problema no poseer un sistema de riego seguro y eficaz, que mejore la utilización del recurso hídrico; ahorrando además costos al contratar menor cantidad de obreros que se dediquen al riego de las plantaciones.

A pesar de que en los últimos años se han desarrollado sistemas para la determinación, automatización y control del riego, son pocos los sistemas que han encontrado en la práctica una aceptación considerable. Esto se debe principalmente a la alta demanda de tiempo, trabajo y capacitación para operar y alimentar de datos e información estos sistemas; además, no se tiene el suficiente conocimiento sobre los resultados que estos tienen sobre el consumo de agua y los rendimientos de los cultivos o plantas, así como los elevados costos de la tecnificación de los sistemas para el control de riego obligando a la mayoría de los productores a depender de la bondad de las lluvias en la época de invierno para la obtención de sus productos (Lugo *et al.*, 2011).

En la actualidad la sistematización de los procesos es una necesidad, convirtiéndose en una herramienta indispensable, por lo que sistematizar el

sistema de riego para el cultivo de la finca “La Lucía” es dar solución a los problemas que se suscitan, tales como el desperdicio del líquido vital porque este no cuenta con un control adecuado del agua y el exceso de esta en el suelo provoca la pérdida del cultivo, por ello la necesidad de implementar un sistema de riego automatizado que brinde una solución, dinámica, confiable y eficiente, con el fin de optimizar la productividad en las cosechas conociendo los requerimientos de riego necesarios para satisfacer la demanda de los cultivos en época de lluvias o en época seca, cuando la producción agrícola depende completamente del agua de riego.

Con tales antecedentes el autor se plantea la siguiente interrogante:

¿Cómo mejorar el sistema de riego en las plantaciones de la finca “La Lucía”?

1.2. JUSTIFICACIÓN

La tecnología se desarrolla a gran escala a través de los años, pero no lo hace sola, a medida que esta evoluciona los seres humanos también, utilizando estas tecnologías con la finalidad de mejorar y automatizar sus procesos.

La investigación, se justifica de acuerdo al reglamento de Tesis de Grado del Manual del Sistema de Investigación Institucional, del Capítulo I Artículo 2 que se enuncia así: “Todo tema de tesis de grado estará relacionado con las líneas de investigación de la carrera del postulante, enmarcado en las áreas y prioridades de investigación establecidas por la ESPAM MFL en concordancia con el Plan Nacional para el Buen Vivir”.

El autor del presente proyecto dando cumplimiento a lo estipulado en el artículo del Reglamento de Tesis de la ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ, se enfoca en cuatro aspectos primordiales, sociales, económicos, legales y ambientales.

Los profesionales necesitan de la sociedad para desarrollarse, por ende su vinculación con ella es beneficiosa, ya que el conocimiento y toda la experiencia adquirida le abrirán muchas puertas en el ámbito profesional y personal, por lo que la universidad se preocupa en formar profesionales de calidad y con calidez humana.

Este proyecto se justifica completamente en el ámbito económico y ambiental por cuanto mejora eficientemente el uso del agua con lo que se beneficiara la producción de los cultivos en esta área. Puesto que con la ejecución de este proyecto no solo se está aportando con la solución si no que sirve de ejemplo a las demás fincas a que implementen este prototipo de sistema que no solo beneficiará a los cultivos sino a la economía y a la seguridad alimentaria de la población en general.

Los sembríos, invernaderos y jardines en la Finca “La Lucia” se beneficiaran con la implementación de la automatización del sistema de riego ayudado por una placa Arduino y actuadores al sistema de irrigación en un área de 100 metros, permitiendo controlar el encendido y apagado de las bombas o válvulas electrónicas que permiten el riego de los cultivos, lo que mejorará el agro y por ende la producción, además del estilo de vida, evitando el desperdicio innecesario del líquido vital, puesto que actualmente su gestión y consumo es objetivo prioritario de organismos internacionales, gobiernos nacionales y de organizaciones no gubernamentales (Miguel *et al*, 2010).

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Implementar un sistema de riego automatizado y tecnificado que permita que los procesos de irrigación se lleven de manera eficaz y eficiente en las áreas de los cultivos agrícolas de la finca La Lucia del Sitio La Pita.

1.3.2. OBJETIVOS ESPECÍFICOS

- Desarrollar los procesos de análisis, diseño y programación de códigos necesarios para la configuración del sistema de riego.
- Realizar prototipos de pruebas con la tarjeta Arduino y verificar el comportamiento de este con los distintos dispositivos a realizar la tesis.
- Implementar la automatización al sistema de riego en el área que se cultiva en la Finca “La Lucía”.

1.4. IDEA A DEFENDER

El sistema de riego automatizado por una placa Arduino permitirá que los procesos de irrigación se lleven de manera eficaz y eficiente en las áreas de los cultivos agrícolas de la finca La “Lucia” del sitio La Pita.

CAPÍTULO II. MARCO TEÓRICO

2.1 CARACTERIZACIÓN DE LA FINCA “LA LUCIA”

La Finca “La Lucia” ubicada en Chone, Canuto específicamente en la Piñuela cuenta con terrenos que son ocupados como potreros y sembríos, los cuales como todo terreno es utilizado para un propósito agrícola el que necesita de un sistema de riego, que cumpla con las necesidades de cada uno de los tipos de procesos en los sembríos y usos que se le da al líquido vital el agua.

2.2. SISTEMA DE RIEGO

Los sistemas de riego ofrecen una serie de ventajas que posibilitan racionalizar el agua disponible. Cualquier sistema de riego debe someterse a un estudio previo para determinar si es el más idóneo, tomando en consideración desde el tipo de vegetación, hasta la forma de distribuir el agua para obtener el mejor rendimiento. Los instrumentos de control de riego: programadores, higrómetros, detectores de lluvia, entre otros, deben distribuirse en función de la orografía, las capacidades hídricas del suelo, las plantaciones García y Briones (2009).

Así mismo expresa García y Briones (2009) que existen muchos y variados sistemas de riego, los cuales se encuentran en permanente revisión, ya que se trata de una tecnología joven que se ha ido desarrollando al mismo tiempo que ha avanzado la sociedad del bienestar. Las zonas verdes han pasado de ser un lujo a una necesidad y el riego es la operación más importante para mantenerlas.

Red General de Riego. Las redes de riego se componen de varios tramos de canalizaciones:

- **Primario.** Va desde el contador hasta las puntas de consumo. Se compone de: bocas de riego, válvulas, electroválvulas y llaves de estaciones.

- **Secundario.** Entre las válvulas, electroválvulas y los mecanismos de distribución del agua: aspersores, difusores y goteros.
- **Distribuidores de agua.** Elementos destinados a distribuir el agua de acuerdo con una pluviometría predeterminada: aspersores, difusores, bocas de riego, goteros, etc.

2.2.1. RIEGO POR ASPERSIÓN

De acuerdo a García y Briones (2009) el riego por aspersión Es aquel sistema de riego que trata de imitar a la lluvia. Es decir, el agua destinada al riego se hace llegar a las plantas por medio de tuberías y mediante unos pulverizadores, llamados aspersores y, gracias a una presión determinada, el agua se eleva para que luego caiga pulverizada o en forma de gotas sobre la superficie que se desea regar.

Para conseguir un buen riego por aspersión son necesarios:

- Presión del agua por medio de una bomba.
- Una planificada red de tuberías, que permita manejar los niveles de presión adecuados al riego.
- Aspersores que permitan abastecer de agua a toda el área de cultivo de una manera eficiente.
- Depósito de agua que permitan conectarse con la red de tuberías

2.2.1.1. RIEGO CON ASPERSORES

El reparto de agua se efectúa de acuerdo con una pluviometría prefijada, y es un sistema idóneo para superficies geoméricamente regulares y de una amplitud considerable. Los principales tipos de aspersores son: Aspersor de impacto y aspersor de turbina. Según la presión de funcionamiento, los aspersores pueden catalogarse en: Baja presión, con presiones de hasta 1.5 kg/cm² y radios de alcance hasta 12 metros; media presión, presiones entre

1,5 y 4,5 kg/cm² y radios de alcance de entre 12 y 25 metros; y alta presión, presiones superiores a 4,5 kg/cm² y radios de alcance de hasta 60 metros. Desde el punto de vista técnico existen otros aspectos que afectan de diversa forma a la idoneidad de un aspersor. Entre estos destacan la uniformidad en la velocidad de rotación, el ángulo y disposición de la tobera o toberas, la altura de la trayectoria, la uniformidad de distribución, el tamaño de las gotas, etc. Respecto al área que los aspersores humedecen se pueden catalogar en: Circulares y sectoriales. Por último, los aspersores se catalogan según sus posiciones de instalación: Aéreos, los que se sitúan sobre la superficie del suelo, y emergentes: los que se instalan enterrados y protegidos por una carcasa (Ambientum, 2008).

2.2.2. INSUMOS DE IRRIGACIÓN

El objetivo de un apropiado manejo de la irrigación es el de maximizar las eficiencias y minimizar los requerimientos de mano de obra y capital para un sistema de riego tanto como sea posible; y, al mismo tiempo mantener un medio ambiente favorable para el crecimiento de la planta en orden de maximizar el rendimiento del cultivo (García y Briones, 2009).

Este método, de uso general, se aplica sobre grandes superficies lanzando un gran volumen de agua controlada y uniforme en forma de lluvia. Por lo general, actúa cubriendo toda el área y es muy adecuado para automatizar la operación. De entrada según Agudelo (2005).

2.3. TECNIFICACIÓN DE PROCESOS DE RIEGO

La tecnificación del riego es la solución a las problemáticas que se suscitan por la falta del líquido vital agua, para tener mejores rendimientos en la producción agrícola por unidad de agua que se es consumida tal como se lo presenta en la Foto 2.1.

Esto obtendrá mejores resultados, condiciones ecológicas. Destinar al cultivo el agua que requiere en cantidad, calidad y oportunidad para optimizar la producción agrícola de los sembríos, es la oportunidad para obtener mejores resultados en calidad y cantidad.



Foto 2. 1 tecnificación de procesos de riego

2.4. DOMÓTICA

La domótica se refiere a la automatización y control encendido / apagado, apertura / cierre y regulación, de aparatos y sistemas de instalaciones eléctricas y electrotécnicas iluminación, climatización, persianas, puertas y ventanas motorizados, el riego, entre otros. de forma centralizada y/o remota. Está determinado por una medición avanzada de la infraestructura de red eléctrica con un medidor inteligente, el cual cuenta con aparatos inteligentes interconectados. Hay una tendencia emergente que ayuda a los consumidores a reducir el consumo de energía de la casa mediante la supervisión y el control de los electrodomésticos y la reprogramación de su tiempo de funcionamiento, de acuerdo con la demanda de energía y de suministro (Vega *et al.*, 2014).

2.4.1. ELECTRÓNICA

Según Vinueza (2015) la electrónica es la rama de la física y especialización de la ingeniería, es la que estudia y emplea sistemas cuyo funcionamiento se basa

en la conducción y el control del flujo microscópico de los electrones u otras partículas cargadas eléctricamente.

Utiliza una gran variedad de conocimientos, materiales y dispositivos, desde los semiconductores hasta las válvulas termoiónicas. El diseño y la gran construcción de circuitos electrónicos para resolver problemas prácticos forman parte de la electrónica y de los campos de la ingeniería electrónica, electromecánica y la informática en el diseño de software para su control. El estudio de nuevos dispositivos semiconductores y su tecnología se suele considerar una rama de la física, más concretamente en la rama de ingeniería de materiales.

La electrónica desarrolla en la actualidad una gran variedad de tareas. Los principales usos de los circuitos electrónicos son el control, el procesado, la distribución de información, la conversión y la distribución de la energía eléctrica. Estos dos usos implican la creación o la detección de campos electromagnéticos y corrientes eléctricas

2.4.2. AUTOMATIZACIÓN

La Real Academia de las Ciencias Físicas y Exactas define la automática como el conjunto de métodos y procedimientos para la sustitución del operario en tareas físicas y mentales previamente programadas. De esta definición original se desprende la definición de la automatización como la aplicación de la automática al control de procesos industriales (Toapanta, 2012).

Por proceso, se entiende aquella parte del sistema en que, a partir de la entrada de material, energía e información, se genera una transformación sujeta a perturbaciones del entorno, que da lugar a la salida de material en forma de producto. Los procesos industriales se conocen como procesos continuos, procesos discretos y procesos batch. Los procesos continuos se caracterizan por la salida del proceso en forma de flujo continuo de material, como por ejemplo la purificación de agua o la generación de electricidad. Los

procesos discretos contemplan la salida del proceso en forma de unidades o número finito de piezas, siendo el ejemplo más relevante la fabricación de automóviles. Finalmente, los procesos batch son aquellos en los que la salida del proceso se lleva a cabo en forma de cantidades o lotes de material (Ponsa y Granollers, 2011).

2.5. PLATAFORMA ARDUINO

2.5.1. SOFTWARE

De acuerdo a Campderrich (2005) Software o soporte lógico de un computador es el conjunto de programas asociados a un computador o a un equipo tecnológico que brinde funciones específicas estos pueden ser de muchos tipos de programación, de control de tratamiento, entre otros.

Conceptualización de Proyectos de Software Libre, Administración de Proyectos de Software Libre y Construcción de Aplicaciones de Software Libre (Álvarez, 2013). Los componentes que se usaron en la parte de software aplicable al Sistema de riego fueron: Lenguaje Arduino SDK (Software Developer Kit) en el cual se escribe, compila los sketches y se cargan al hardware.

2.5.1.1. LENGUAJE ARDUINO (SDK)

El entorno del lenguaje de programación de Arduino de acuerdo con Pomares (2009). Es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados. Pensando en los profesores, Arduino está basado en el entorno de programación de Processing con lo que el estudiante que aprenda a programar en este entorno se sentirá familiarizado con el entorno de desarrollo Arduino.

2.5.1.2. SOFTWARE AMPLIABLE Y DE CÓDIGO ABIERTO

El software Arduino está publicado bajo una licencia libre y preparada para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente código en AVR C en los programas (Arduino, 2012).

2.5.2. HARDWARE

El hardware son componentes físicos de equipos tecnológicos, industriales esto quiere decir todo lo que puede tocarse o palpar con la mano según Díaz (2012). Es la parte tangible de una computadora (Díaz, 2012).

2.5.2.1. HARDWARE AMPLIABLE Y DE CÓDIGO ABIERTO

Conforme Herrador (2009) Arduino está basado en los micros controladores ATMEGA168, ATMEGA328y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de desarrollo para entender cómo funciona y ahorrar algo de dinero.

2.5.3. ARDUINO

Arduino es una plataforma de prototipos electrónica de código abierto (open source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como trabajo, hobby y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros

artefactos así lo define Evans (2010). El micro controlador de la placa se programa usando el Arduino Programming Lenguaje (basado en Wiring1) y el Arduino Development Environment (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador, las placas se pueden ensamblar a mano o encargarse pre ensambladas.

2.5.3.1. PLACA ARDUINO

Arduino es una herramienta para hacer que los ordenadores puedan sentir y controlar el mundo físico a través de un ordenador personal. Es una plataforma de desarrollo de computación física (physical computing) de código abierto, basada en una placa con un sencillo micro controlador y un entorno de desarrollo para crear software (programas) para la placa (Pomares, 2009).

2.5.3.2. CARACTERÍSTICAS DE LA PLACA ARDUINO

Las placas han ido evolucionando como su software, al inicio las primeras placas utilizaban un chip FTDI “FT232RL” para comunicarse por puerto USB al computador y un procesador para ser programado, luego se utilizó un microcontrolador especial para cumplir esta función como en el caso de Arduino “uno”, que tenían un micro para ser programado y otro para la comunicación, en la actualidad se usa un único microcontrolador que se compromete en llevar a cabo la comunicación y sobre el que también se descargan las instrucciones a ejecutar (Arduino, 2011).

2.5.3.3. DESCRIPCIÓN DE LA PLACA ARDUINO UNO R3

La placa Arduino uno contiene diferentes componentes esenciales, para el funcionamiento de esta con distintos componentes, detallados en la figura 2.1.

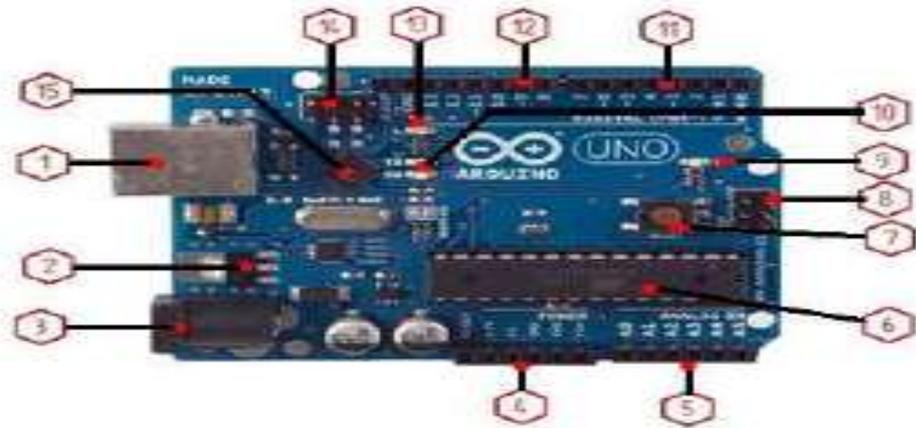


Figura 2. 1 Características de la placa Arduino

1. CONECTOR USB

Proporciona la comunicación para la programación y la toma de datos, también provee una fuente de 5v para alimentar el Arduino pero de baja corriente por lo que no sirve para alimentar motores de gran potencia.

2. REGULADOR DE VOLTAJE DE 5v

Se encarga de convertir el voltaje ingresado por el plug 3, en un voltaje de 5v regulado necesario para el funcionamiento de la placa y para alimentar circuitos externos.

3. PLUG DE CONEXIÓN PARA FUENTE DE ALIMENTACIÓN EXTERNA

Es el voltaje que se suministra que debe ser directo y estar entre 6v y 18v o hasta generalmente 20v, se debe tener cuidado de que el terminal del centro de plug quede conectado a positivo ya que algunos adaptadores traen la opción de intercambiar la polaridad de los cables.

4. PUERTOS DE CONEXIÓN

Es constituido por 6 pines de conexión con las funciones de RESET que permite resetear el micro controlador al enviarle un cero lógico. Pin 3.3V provee de una fuente de 3.3VDC para conectar dispositivos externos como en la protoboard por ejemplo. Pin 5V es una fuente de 5VDC para conectar dispositivos externos

5. PUERTOS DE ENTRADAS ANÁLOGAS

Lugar donde se conectan las salidas de los sensores análogos. Estos pines solo funcionan como entradas recibiendo voltajes entre cero y cinco voltios directos.

6. MICRO CONTROLADOR ATMEGA 328

Implementado con los Arduino uno y Arduino Mega en la versión SMD del Arduino uno R2, se usa el mismo micro controlador pero en montaje superficial, en este caso las únicas ventajas son la reducción del peso y ganar un poco de espacio.

7. BOTÓN RESET

Permite resetear el micro controlador haciendo que reinicie el programa.

8. PINES DE PROGRAMACIÓN ICSP

Son usados para programar micro controladores en protoboard o sobre circuitos impresos sin tener que retirarlos de su sitio.

9. LED ON

Enciende cuando el Arduino está conectado o encendido con el voltaje correcto.

10. LEDS DE RECEPCIÓN Y TRANSMISIÓN

Se encienden cuando la tarjeta se comunica con el PC. El Tx indica transmisión de datos y el Rx recepción.

11. PUERTOS DE CONEXIONES DE PINES DE ENTRADAS O SALIDAS DIGITALES

La configuración como entrada o salida debe ser incluida en el programa. Cuando se usa la terminal serial es conveniente no utilizar los pines cero (Rx) y uno (Tx). Los pines 3, 5 y 6 están precedidos por el símbolo ~, lo que indica que permiten su uso como salidas controladas por ancho de pulso PWM.

12. PUERTO DE CONEXIONES 5 ENTRADAS O SALIDAS

Las salidas 9, 10 y 11 permiten control por ancho de pulso; la salida 13 es un poco diferente pues tiene conectada una resistencia en serie, lo que permite conectar un led directamente entre ella y tierra. Finalmente hay una salida a tierra GND y un pin AREF que permite ser empleado como referencia para las entradas análogas.

13. LED PIN 13

Indica el estado en que se encuentra.

14. PINES DE PROGRAMACIÓN ICSP

Son usados para programar micro controladores en protoboard o sobre circuitos impresos sin tener que retirarlos de su sitio.

15. CHIP DE COMUNICACIÓN

Permite la conversión de serial a USB sus partes y características se muestran en la figura 2.2.

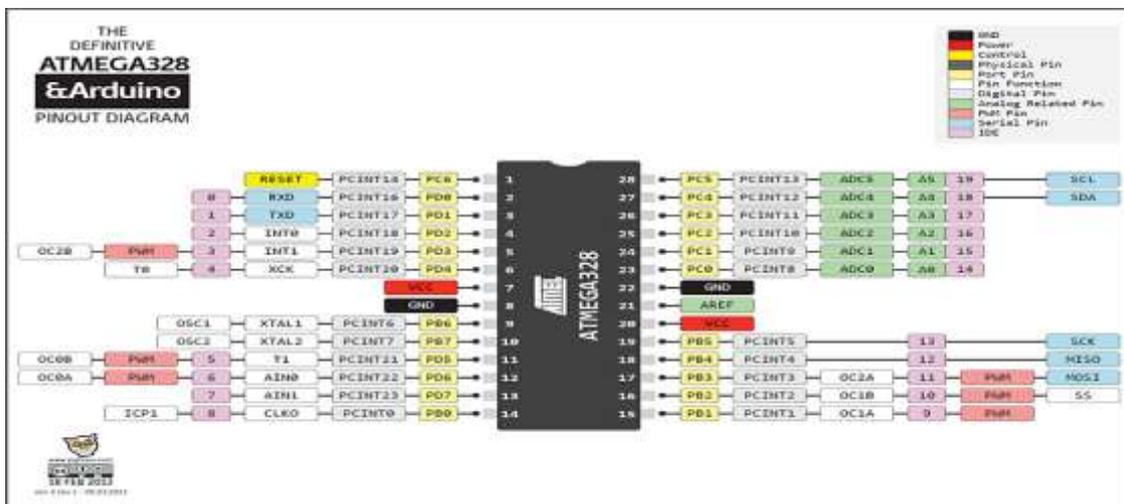


Figura 2. 2 Placa Arduino Chip de comunicación

2.5.3.4. DESCRIPCIÓN DE LA PLACA ARDUINO MEGA 2560

El Arduino mega 2560 visto en la figura 2.3. es una placa electrónica basada en el atmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas pwm), 16 entradas analógicas, 4 uarts

(hardware puertos serie), un oscilador de cristal de 16 mhz, una conexión usb, un conector de alimentación, un header icsp, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; simplemente conectarlo a un ordenador con un cable USB o el cable de poder con un adaptador para la batería. La placa mega es compatible con la mayoría de los escudos diseñados para el Arduino, su resumen pertinente a los voltajes que maneja esta placa se muestran en la tabla 2.1.

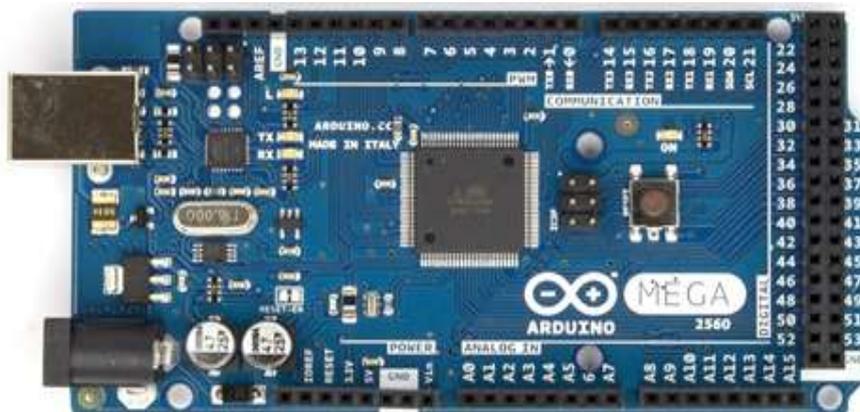


Figura 2. 3 Placa Arduino Mega 2560

Resumen

| | |
|----------------------------------|---|
| Microcontroladores | Atmega2560 |
| Tensión De Funcionamiento | 5V |
| Voltaje de entrada (recomendado) | 7-12V |
| Voltaje de entrada (límites) | 6-20V |
| Digital pines I / O | 54 (de las cuales 15 proporcionan salida PWM) |
| Botones de entrada analógica | 16 |
| Corriente DC por E / S Pin | 40 mA |
| Corriente DC de 3.3V Pin | 50 mA |
| Memoria Flash | 256 KB de los cuales 8 KB utilizado por el gestor de arranque |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Velocidad De Reloj | 16 MHz |

Tabla 2. 1 Resumen descripción placa Arduino

2.5.3.5. SEÑALES DIGITALES Y SEÑALES ANALÓGICAS

Las señales se clasifican en digitales y analógicas una de las clasificaciones es identificar o distinguir las señales analógicas y señales digitales.

Según Torrente (2013) las señales digitales son aquellas que tienen un número finito de valores posibles que se suelen llamar valores discretos por ejemplo si se consideramos como señal el color emitido por un semáforo, es fácil ver que esta es tipo digital, porque solo puede tener tres valores concretos, diferenciados y sin posibilidad de transición progresiva entre otro; rojo, ámbar y verde.

Señales analógicas es aquella que tiene infinitos valores posibles dentro de un rango determinado a lo que se suele llamar “tener valores continuos” la mayoría de magnitudes físicas temperaturas, sonido, luz, son analógicas

2.6. COMPONENTES A TRABAJAR CON PLACA ARDUINO

A continuación se detallan varios de los componentes que se usan para trabajar con la placa Arduino:

2.6.1. ARDUINO PANTALLA LCD



Figura 2. 4 Pantalla Lcd para Arduino 16x2 Fuente Arduino (2013)

La pantalla Lcd vista en la figura 2.4 es un dispositivo diseñado para mostrar información en forma gráfica. Lcd significa Liquid Crystal Display (Display de cristal líquido). La mayoría de las pantallas Lcd vienen unidas a una placa de

circuito y poseen pines de entrada/salida de datos. Como se podrán imaginar, Arduino es capaz

Estas pantallas constan de 16 pines. De izquierda a derecha, sus usos son los siguientes:

- **Pin 1** – VSS o GND
- **Pin 2** – VDD o alimentación (+5V)
- **Pin 3** – Voltaje de contraste. Se conecta a un potenciómetro.
- **Pin 4** – Selección de registro. Aquí se selecciona el dispositivo para su uso.
- **Pin 5** – Lectura/Escritura. Dependiendo del estado (HIGH o LOW), se podrá escribir o leer datos en el Lcd.
- **Pin 6** – Enable. Es el pin que habilita o deshabilita el Lcd.
- **Pin 7 hasta Pin 14** – Son los pines de datos por donde se envía o recibe información.
- **Pin 15** – El ánodo del Led de iluminación de fondo (+5v).
- **Pin 16** – El cátodo del Led de iluminación de fondo (GND).

2.6.2. RTC ds1307 ARDUINO

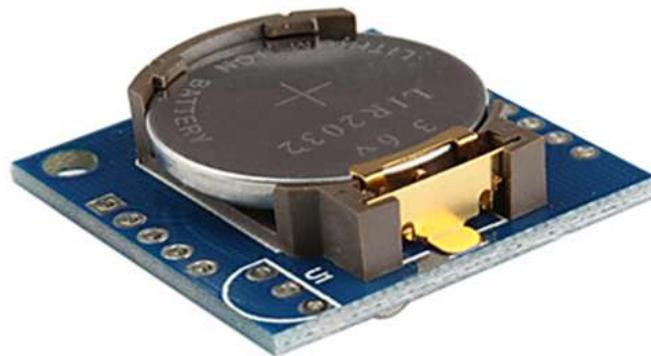


Figura 2. 5 Rtc ds1307 modulo para Arduino

En la figura 2.5 se muestra algo parecido a un reloj de computadora, (generalmente en forma de circuito integrado) que mantiene la hora actual. Los Rtc están presentes en casi todos los dispositivos electrónicos que necesitan

del tiempo actual. No debe confundirse con la señal de reloj medidas en frecuencia que rige la electrónica digital.

2.6.3. RELÉ PARA ARDUINO.



Figura 2. 6 Relé 2 canales 5v Arduino

Según Monk (2012) la tarjeta de relé opto acoplada, incluye 2 canales para ser controlados en forma remota. Ideal para controlar dispositivos en el hogar o en la industria. Cada canal es controlado por una entrada Ttl, la cual puede ser fácilmente controlada por un micro controlador o Arduino. Esta placa mostrada en la figura 2.6 requiere de una alimentación de 5v.

Características:

- 2 canales independientes protegidos con opto acopladores
- 2 Relés (Relays) de 1 polo 2 tiros
- El voltaje de la bobina del relé es de 5v
- Led indicador para cada canal (enciende cuando la bobina del relé esta activa)
- Activado mediante corriente: el circuito de control debe proveer una corriente de 15 a 20 mA
- Puede controlado directamente por circuito lógicos
- Terminales de conexión de tornillo
- Terminales de entrada de señal lógica.

2.6.4. TARJETAS SD A 3.3v.

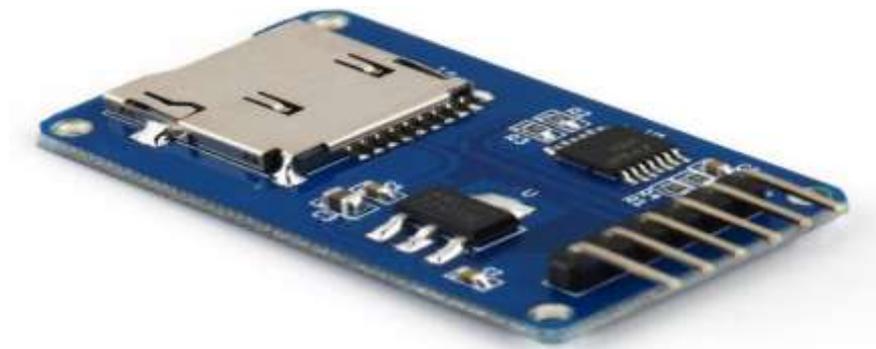


Figura 2. 7 Modulo lector de tarjeta Sd Arduino

Tarjetas Sd sólo funcionan a 3.3v y tanto el poder y la E / S de los niveles deben ser acomodados. El módulo se muestra aquí utiliza FET para cambio de nivel y un regulador de 3.3v por el poder cuando se opera desde 5.0v. Un interruptor permite que el módulo que se utilizará con Arduino. El módulo de la tarjeta Sd fijado en 3.3v funciona bien El módulo tiene una toma de tarjeta Micro SD en la parte trasera, la misma que ha sido probada por el autor con tarjetas de varios tamaños, disponibles en el mercado. Trabajo bien con la biblioteca Sd FAT. Estas tarjetas tienen el formato FAT32 y SD figura 2.7.

Módulo Pin

10 (SS) para CS

11 (MOSI) de DI

12 (MISO) HACER

13 (SCK) a CLK

G a GND y + 5v

2.6.5. MICRO CONTROLADORES



Figura 2. 8 Micros controladores

Un micro controlador combina los recursos básicos de una unidad central de procesamiento (CPU), la memoria y los recursos de entrada y salida, en un único circuito integrado (Pallas, 2007).

Actualmente los micros controladores mostrados en la figura 2.8 son un estándar para aplicaciones de mediana complejidad, por su facilidad de manejo y nivel de prestaciones.

Sus características principales son:

- Capacidad de procesos de palabras de 8 bits.
- Frecuencia de reloj de 30 Mhz. (o más).
- Múltiples puertos de entrada/ salida programables.
- Instrucciones orientadas al proceso de señales bit a bit.
- Circuito de reloj incorporado.
- Bajo consumo en versiones CMOS.
- Comparadores de tensión.
- Temporizadores.

2.6.6. ELECTROVÁLVULA



Figura 2. 9 Válvula eléctrica

Válvula eléctrica vista en la figura 2.9 usada para el control automático de líquido y gas está compuesta por un circuito que cierre y abre dicha válvula (Méndez, y Gómez, 2010).

2.6.7. MICROPROCESADOR

Es por excelencia, el corazón del ordenador según Álvarez *et al.* (2009) consiste en un chip o sistema electrónico integrado capaz de interpretar y ejecutar órdenes. Su función es la de realizar operaciones y cálculos, tanto de tipo aritmético como lógico, sobre datos que le envían los programas, controlando y dirigiendo todo el proceso de datos. Consta de unas pastillas de conexión insertado en una placa base del ordenador, una base de plástico, cristal y silicio, equivalente a millones de componentes electrónicos (transistores diodos y capacitores).

El microprocesador accede a la información, la procesa y da la respuesta adecuada. Así, la mecánica básica de un ordenador podría decirse.

El microprocesador, para llevar a cabo su función primordial (procesar datos), se divide en tres partes, cada una de las cuales tiene una tarea determinada:

- Unidad de control: es la parte del microprocesador encargada de interpretar las instrucciones que le llegan de la memoria RAM.
- Unidad aritmética lógica (VAL, más comúnmente, ALU); su función es la de realizar operaciones matemáticas y lógicas que le indica la Unidad de Control.
- Registros de Almacenamiento: esta parte permite almacenar los resultados de los cálculos por la ALU, además de la dirección de memoria donde se encuentra la siguiente.

2.7. CIRCUITOS IMPRESOS

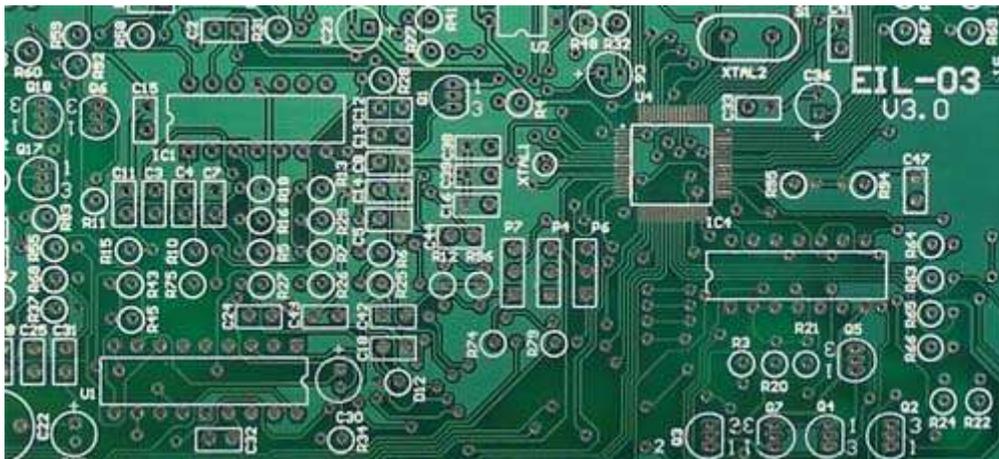


Figura 2. 10 Circuito Impreso

Un circuito impreso o PCB en inglés, es una tarjeta o placa utilizada para realizar el emplazamiento de los distintos elementos que conforman el circuito y las interconexiones eléctricas entre ellos vista en la figura 2.10.

Antiguamente era habitual la fabricación de circuitos impresos para el diseño de sistemas mediante técnicas caseras, sin embargo esta práctica ha ido disminuyendo con el tiempo. En los últimos años el tamaño de las componentes electrónicas se ha reducido en forma considerable, lo que implica

menor separación entre pines para circuitos integrados de alta densidad. Teniendo también en consideración las actuales frecuencias de operación de los dispositivos, es necesaria una muy buena precisión en el proceso de impresión de la placa con la finalidad de garantizar tolerancias mínimas (Páez *et al.*, 2007).

2.8. MÉTODOS Y TÉCNICAS

2.8.1. TÉCNICA DE LA ENCUESTA

La encuesta permite un panorama amplio de la situación, brindando una visión histórica, así como comparativa con el entorno se utiliza con el fin de obtener información de la población de una determinada zona sobre fenómenos subjetivos (Galavis *et al.*, 2010).

2.8.2. METODOLOGÍA HARDWARE LIBRE

La metodología posee tres procesos mostrados en las figuras 2.11 – 2.17 en el de conceptualización se busca delimitar los alcances que se quiere para el proyecto en estudio, en el proceso de administración se busca la planificación para el diseño, fabricación y pruebas del dispositivo. Por último el proceso de desarrollo en el cual se especifican los pasos que en principio se deben cumplir, dependiendo de la naturaleza del dispositivo (Medrano, 2011).



Figura 2. 11 El Proceso de desarrollo Fuente: Medrano (2011)

2.8.2.1. PROCESO DE CONCEPTUALIZACIÓN DE PROYECTOS

En este proceso se analizan problemas y necesidades de las comunidades que pudiesen requerir de una solución en área de hardware. El análisis planteado conlleva a la reflexión sobre los problemas y sus posibles soluciones. La actividad de reflexión tiene como objetivo principal proponer soluciones pertinentes a los problemas planteados, en las cuales se consideren tanto los beneficios como el impacto que dichas soluciones puedan causar sobre la comunidad. En este proceso se debe destacar, que las soluciones planteadas o parte de ellas sean pertinencia de otra área como por ejemplo el desarrollo de alguna aplicación de software requerido para el diseño del hardware.

- Actividad: nombre específico de la actividad a desarrollar dentro del proceso de administración.
- Responsable/Participantes: describe los individuos que tienen asignada una responsabilidad o participan en alguna actividad.
- Insumo: Entrada (documento, plantilla, informe, etc.) necesaria para el desarrollo de una actividad específica.
- Observaciones: Campo para establecer observaciones relacionadas a la actividad.

- Técnicas/Herramientas/Plantillas: listado de técnicas, herramientas, plantillas que pueden ser aplicables para desarrollar una actividad específica.
- Productos: Listado de productos finales resultado de una actividad específica

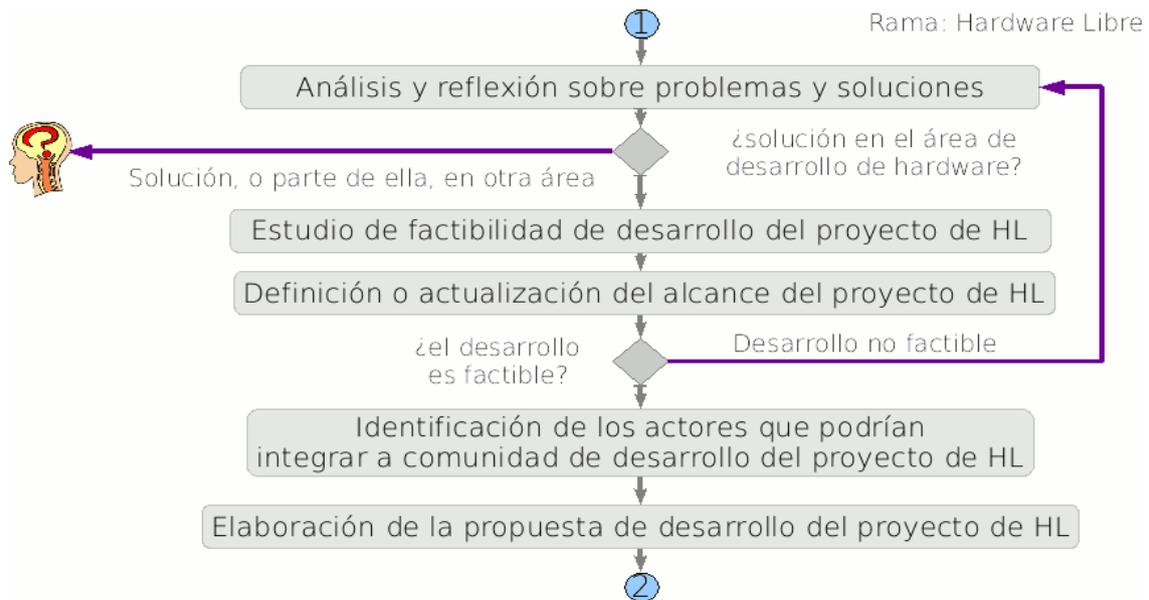


Figura 2. 12 Proceso detallado de conceptualización Fuente: Medrano (2011)

2.8.2.2. PROCESO DE ADMINISTRACIÓN DE PROYECTOS DE HL

El proceso de administración de la Metodología de Desarrollo de Hardware Libre comprende un conjunto de actividades para coordinar y mantener el orden de un proyecto de desarrollo de hardware libre. Estas actividades estarán orientadas a facilitar lo planteado en el proceso de conceptualización.

El proceso de administración requiere que se establezca el rol en uno de los integrantes del equipo como Coordinador del proyecto de desarrollo de hardware.

El Coordinador debe velar por el seguimiento y cumplimiento de las actividades de desarrollo, promover una comunidad de desarrollo y colaboración en torno

al proyecto, la cual será la encargada de elaborar el plan del proyecto de desarrollo de hardware.

Para ilustrar las actividades propuestas en el proceso de administración se muestra un diagrama de flujo.

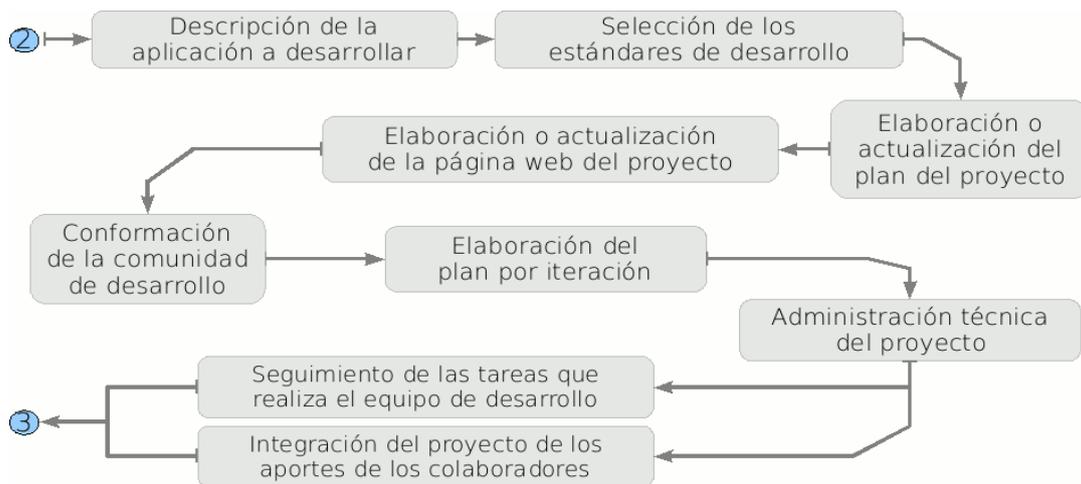


Figura 2. 13 Proceso de administración procesos de desarrollo de hardware libre

2.8.2.3. PROCESO DE DESARROLLO DE PROYECTOS EN HL

A continuación se muestra un diagrama general del proceso de desarrollo de proyectos de hardware libre.



Figura 2. 14 Proceso de Desarrollo de Proyectos en Hardware Libre

Se parte de una descripción detallada del alcance y características del hardware a desarrollar, descripción que ha sido preparada en los procesos de conceptualización y administración.

Al comienzo del proceso de desarrollo dependiendo de la naturaleza del hardware a diseñar, se puede dividir en tres pasos concurrentes: Especificación de Hardware Estático (a), Programación de Dispositivos (b), Desarrollo de IC (c). Esas áreas pueden activarse o no según los requerimientos del proyecto. En todo caso, normalmente siempre estará incluida en alguna medida la Especificación de Hardware Estático. Estos pasos de desarrollo se ocupan de generar y depurar los diseños que sean necesarios para implementar las características requeridas. Este proceso necesariamente las lleva a trabajar en forma coordinada, para que sus resultados puedan integrarse entre sí.

Luego los pasos (a), (b) y (c), concurren en la etapa de integración, en donde se ajustan todos los detalles necesarios para obtener un diseño completo del hardware. En esta etapa puede detectarse la necesidad de reformular el alcance y características del proyecto, debido a las posibles incompatibilidades entre los diseños y configurar los protocolos de comunicación.

Una vez obtenido el diseño integrado, puede entonces someterse al mismo a verificaciones que permitan depurarlo en su conjunto, lo cual es realizado, posiblemente mediante simulaciones, en la siguiente etapa. Los resultados de estas verificaciones pueden dar pie a modificaciones en el proceso de integración o en la formulación del alcance y características del proyecto.

Luego, se procede a la fabricación del prototipo, el cual es luego sometido a diversos protocolos de pruebas. Los resultados de esas pruebas pueden revelar la necesidad de realizar modificación en cualquiera de las etapas anteriores del proyecto.

La liberación de los diseños del hardware, se ha considerado de varias formas, liberación de versiones preliminares llamadas de prueba que se pueden obtener en cualquier paso del proceso de desarrollo (figura 4). Las versiones

estables sólo pueden ser liberadas en cualquiera de los cuatro últimos bloques mientras las versiones de prueba pueden ser liberadas en cualquier momento del ciclo de desarrollo. Estas liberaciones proporcionan mejoras que pueden ser compartidas con los desarrolladores y colaboradores para la obtención del dispositivo final.

A continuación las especificaciones de Hardware Estático, programación de dispositivos, desarrollo de IC:

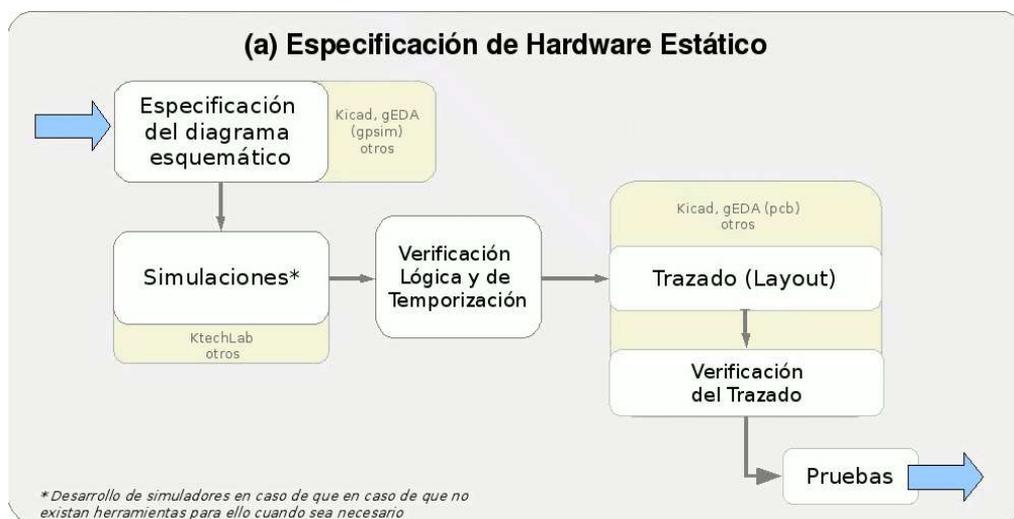


Figura 2. 15 Especificación de Hardware Estático

Seguidamente las especificaciones de la programación de dispositivos.



Figura 2. 16 Programación de dispositivos

Programación de Dispositivos

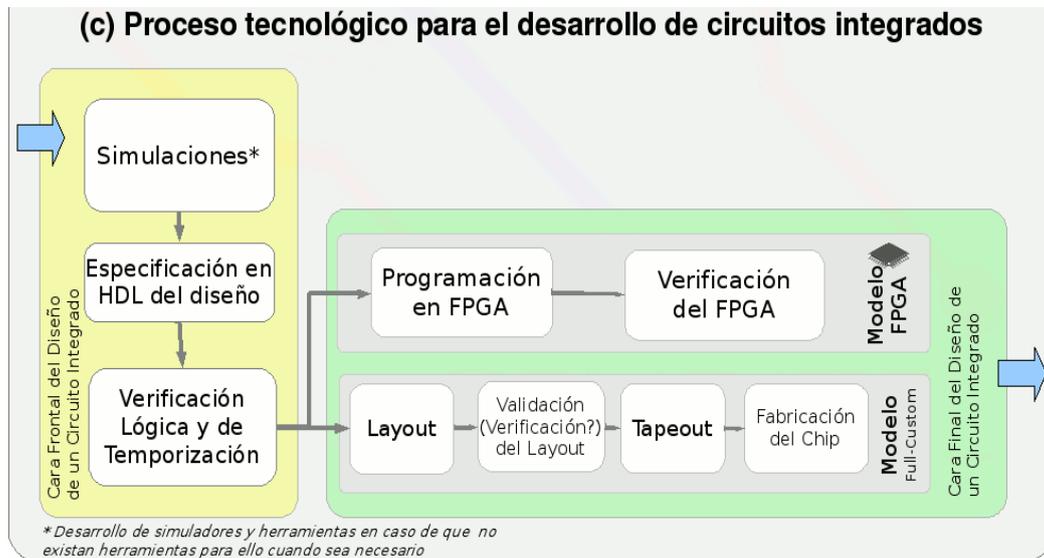


Figura 2. 17 Desarrollo de circuitos

CAPÍTULO III. DESARROLLO METODOLÓGICO

El autor realizó la automatización y control del sistema de riego con una placa Arduino y actuadores en la Finca “La Lucia” del sitio La Pita - Parroquia Canuto - Cantón Chone, vía a la ESPAM MFL. Para el efecto se utilizaron diversos métodos y técnicas como la Investigación de campo, tecnificación y Metodología Hardware Libre.

3.1. MÉTODOS Y TÉCNICAS

La principal idea de estos métodos fue estudiar un fenómeno para un fin, en dicho proyecto el fin fue mejorar el control de riego, reducir el consumo de agua, bajar costos y mejorar la productividad en las plantaciones de la finca “La Lucia”.

3.1.1. MÉTODO DEDUCTIVO

El autor utilizó este método con el fin de argumentar los procesos de los análisis y recopilación de datos sobre las técnicas que actualmente se llevan en la Finca “La Lucia” sobre el sistema de riego implementado en las ares de sembrío. Lo que ha permitido una mejor comprensión en la planeación de soluciones para la mejora del control de riego.

3.1.2. TÉCNICA DE LA ENCUESTA

En el inicio de esta investigación se utilizó esta técnica durante las conversaciones que se mantuvieron con los propietarios de la finca “La Lucia”, esta encuesta permitió al autor conocer a fondo las condiciones del sistema de riego las necesidades, requerimientos sobre cada uno de los procesos que se llevan en el riego.

3.1.3. TECNIFICACIÓN

El autor del proyecto de automatización del sistema de riego utilizó la tecnificación como solución a la problemática que se suscitaba por la falta del control adecuado del líquido vital agua, para mejoras del rendimiento en la producción agrícola por unidad de agua que se es consumida.

Los componentes usados en la fase de desarrollo son: Placa Arduino mega, son: display led (12x2), Micro Sd card adapter, Micro Sd, módulo Rtc, módulo relé, electroválvula, resistencia, potenciómetros, diodos led, baterías, protoboard, cables de conexión hembra y macho, cautín, estaño, multímetro, pinzas, micro controladores, teclado matricial 4x3, circuito impreso.

3.2. MÉTODO INFORMÁTICO

3.2.1. MÉTODO HARDWARE LIBRE

Tal proyecto se lo ejecutó en un tiempo de seis meses, donde se desarrolló la implementación de un sistema de riego con sus respectivos actuadores, todo esto detallado a continuación en los diferentes procesos de la metodología empleada como lo es la Metodología hardware libre.

3.2.2. PROCESOS DE CONCEPTUALIZACIÓN

En el proceso de conceptualización se obtuvo el análisis completo sobre las necesidades y funciones que presentaba el sistema de riego. El autor observó el funcionamiento y desempeño que el sistema mostraba en los cultivos, con tales antecedentes se procedió a realizar una encuesta a los habitantes de la finca la Lucia donde se obtuvo la información para elaborar una encuesta (ANEXO 1), la cual ayudó a definir el alcance que debía tener la automatización

propuesta en la Finca “La Lucia” respecto al sistema de riego en base a las necesidades hídricas del cultivo.

En los cuadros 3.1 - 3.3 se detallan las fases del proceso de conceptualización.

Cuadro 3.1 Análisis y reflexión sobre problemas y soluciones

| ANÁLISIS Y REFLEXIÓN SOBRE PROBLEMAS Y SOLUCIONES | | | | | |
|--|---|----------------------------|---|---|--|
| Actividad | Responsables Participantes | Insumo | Observaciones | Técnicas/Herramientas | Productos |
| Identificar problemas y necesidades | Responsable: El autor. Participantes: Habitantes | Problemas y requerimientos | | Técnicas: Encuesta | Encuesta Ver Anexo(I) |
| Análisis y reflexión sobre los problemas y sus posibles soluciones | Responsable: El autor. Participantes: Habitantes | | Se consideró la solución más adecuada en base a la necesidad de mejorar el control de riego | Técnicas: Planificación estratégica de la ubicación del control y módulos | Realizar una automatización del control de riego |

CUADRO 3.2 Definición del alcance de la investigación

| DEFINICIÓN DEL ALCANCE DE LA INVESTIGACIÓN | | | | | |
|---|--|---|----------------------|------------------------------|--|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/Herramientas | Productos |
| Definición del alcance de la investigación del proyecto | Responsable: El autor. Participantes: Habitantes El autor. | Propuesta del desarrollo de la solución y su factibilidad | | Alcance del proyecto | Identificar las necesidades del procesos de sistema de riego |

Cuadro 3.3 Elaboración de la propuesta de desarrollo del proyecto

| ELABORACIÓN DE LA PROPUESTA DE DESARROLLO DEL PROYECTO. | | | | | |
|---|--|---|---------------|-----------------------------------|--|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/ Herramientas | Productos |
| Elaborar la propuesta del desarrollo del Proyecto | Responsable: El autor. Participantes: Habitantes El autor. | Automatización de los proceso de riego en el área de sembrío de la finca "La Lucia" | | Hardware Libre/ Automatización | Ficha técnica aportada por el autor del proyecto ver Anexo(II) |

3.2.3. PROCESO DE ADMINISTRACIÓN

El proceso de administración se elaboró con base a los resultados que se extrajo de los distintos procesos de conceptualización, se tomó decisiones acerca del sistema en cuanto a la arquitectura, necesidades del sistema y la asignación de los componentes de hardware y software por lo que el autor optó por utilizar la ficha técnica realizada en la fase anterior, con los requisitos, propósito, alcance, detalles y restricciones mostrados en los cuadros 3.4 - 3.6.

Cuadro 3. 4 Descripción del dispositivo a desarrollar.

| DESCRIPCIÓN DEL DISPOSITIVO A DESARROLLAR. | | | | | |
|--|--|---------------------------|--|--|--|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/ Herramientas/ Plantillas | Productos |
| Automatización de los procesos del control de riego con una placa Arduino y actuadores | Responsables: El autor. Participante: El autor. | Ficha técnica ver Anexo 1 | El usuario modifique el horario de riego de acuerdo a la conveniencia de cada sembrío. | Descripción del dispositivo requerimientos | Esquema y Diagrama electrónico Ver figura 4.4. |

Elaboración o actualización del plan del proyecto.

Cuadro 3. 5 Elaboración o actualización del plan del proyecto.

| ELABORACIÓN O ACTUALIZACIÓN DEL PLAN DEL PROYECTO. | | | | | |
|--|--|--|--|---|---|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/ Herramientas/ Plantillas | Productos |
| Automatización de los procesos del control de riego con una placa Arduino y actuadores | Responsables: El autor. Participante: El autor. Habitantes | Mejorar los procesos de control de riego, Automatizar el proceso de riego en el área de sembrío de la finca "La Lucia" | Se recomienda que el autor del proyecto participe con el desarrollo de hardware de acuerdo a las necesidades de la problemática. | Experimentación. | Controlar a horas adecuadas el control de riego. Ver Anexo IV |

Cuadro 3. 6 Administración técnica del proyecto

| ADMINISTRACIÓN TÉCNICA DEL PROYECTO | | | | | |
|---|---|-----------------------|--|---|--|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/ Herramientas/ Plantillas | Productos |
| Requerimientos del código fuente sistema de Riego | Responsable: El autor. Participante: El autor. Habitantes | Arduino-1.6.4-windows | Uso de librerías adecuadas a los módulos del Arduino | Caso de uso | Diagrama de flujo para elaboración del código fuente. Ver figura 4.1 |

PROCESO DE ADMINISTRACIÓN DE HARDWARE Y SOFTWARE

El proceso de administración de hardware se empleó el componente Arduino como hardware libre, hardware programable, conveniente a las necesidades que presenta el control de riego, adaptable a múltiples componentes periféricos, que son necesarios para complementar la automatización del control de riego, se defino los procesos que cumplirán cada uno de los módulos y actuadores, de acuerdo a la información acerca de las características y desempeño necesarias, se procedió a la búsqueda de los componentes a implementar en la automatización del sistema de riego.

DISEÑO DE OBJETOS

Se elaboró un modelo detallado del sistema de riego, sin descender en detalles particulares, para elaborar los procesos que forman parte en el control del riego con enfoque a la orientación del mundo real.

En la figura 3.2 Se observa el diseño terminado de la placa Arduino con sus respectivos actuadores en el control del sistema de riego, el que se integró a los conductos que distribuirán el agua por los puntos o aspersores establecidos en el área de riego, el sistema trabaja junto a una electroválvula que es controlada por el sistema de riego de acuerdo al horario programado por el usuario.

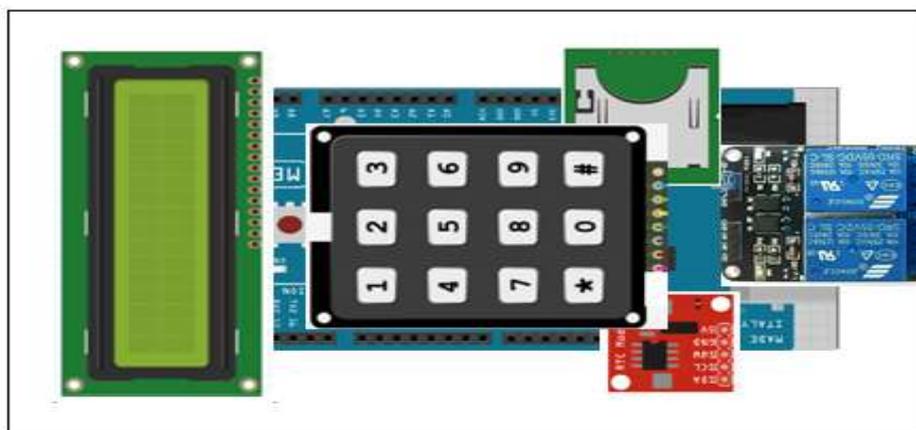


Figura 3. 1 Diseño del control de riego y sus actuadores

En este proceso de diseño se determinó el alcance de la implementación de automatizar el sistema de riego, el cual se dividió el área en diferentes secciones en el que se implementó, actuadores de riego y puntos de conexiones, esto se lo efectuó con la ayuda del texto guía llamado Hunter Industries Incorporate.

3.2.4. PROCESO DESARROLLO

El autor dividió este proceso en tres etapas como lo refiere el proceso de desarrollo, mismas que son: Especificación de Hardware, Programación de dispositivos y Desarrollo que se demuestran en el cuadro 3.7.

Cuadro 3.7 Especificación de hardware

| ESPECIFICACIÓN DE HARDWARE ESTÁTICO | | | | | |
|--|---|---|---|------------------------------|--|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/Herramientas | Productos |
| Especificación del diagrama esquemático | Responsables: El autor. Participantes: El autor. | Especificaciones del diseño esquemático | Acondicionamiento de las posiciones del módulo según su estructura. | Diseño esquemático | Esquema de montaje electrónico Ver Figura 3.2 |
| Verificación del Trazado | Responsables: El autor. Participantes: El autor. | Circuito impreso en digital | | Fritzing.0.9.2b. 64 bits | Impresión de diseño del circuito impreso |

-ESPECIFICACIÓN DE HARDWARE

Con la información que suministra la página web de Arduino el autor analizó las funciones de la placa, en las que encontró que la placa Arduino uno r3 y la placa Arduino mega eran las más adecuadas para el control de riego, procediendo a adquirir las placas comprobando el desempeño y función en la

adaptación de los diferentes módulos a utilizar descritos a continuación y mostrado en el Cuadro 3.8.

En principio las placas presentaron las mismas ventajas, sin embargo al momento de analizar los diferentes módulos y actuadores de las dos placas se encontró con la problemática que la placa Arduino r3 consta con 13 puertos digitales limitando la inclusión de actuadores, procediendo a utilizar la placa Arduino mega ya que cuenta con 58 puertos digitales de entradas y salidas capaces de soportar mayores módulos y actuadores.

Modulo lector micro Sd. adapter V1.0 el autor analizó el módulo lector de micro Sd. en lo que refiere a las características, requerimientos y funcionalidades necesarias al control de riego, la dimensión que presenta el módulo permitió la implementación en la placa Arduino, en cuanto a lo electrónico el dispositivo no necesitó de ningún convertidor de 5v a 3.3v que en versiones anteriores se necesitaba para funcionar adecuadamente con los voltajes internos del Arduino mega, ya que este lo incorpora; Para el almacenamiento de los reportes que genera el control de riego se analizó, los tipos de memorias en cuanto a la compatibilidad que acepta el módulo lector micro Sd, se obtuvo que lo recomendado era incorporar una tarjeta micro Sd. de 2Gb, la que permitió revisar los reportes generados desde un teléfono móvil, Pc o Tablet lo que permitirá al usuario mejorar el horario en el control de riego.

La visualización para el usuario sobre el desarrollo que efectúa el control de riego en los procesos se presentara por medio de un **módulo Lcd**, el autor analizó varios modelos de módulos Lcd, en base a los requerimientos, características, funcionalidades y desempeño, se observó que todos los módulos de este tipo presentan la misma funcionalidad por lo que se concentró en indagar sobre sus dimensiones ya que al implementarse en la placa, debe proporcionar espacio a los diferentes dispositivos y módulos que conlleva, dando como resultado la elección del módulo Lcd, 2x16 como pantalla del control de riego.

Módulo Rtc, para mantener el control de riego en fecha y hora exacta se optó por instalar un módulo Rtc, el autor analizó las características y desempeño, tal como la batería que usa y el voltaje de ingreso que necesitó para su función, en las dimensiones y funciones son similares a otros módulos, se optó por el módulo Rtc 12c, este trae integrado un convertidor de voltaje de 5v a 3.3v que permite manejar los voltajes internos del Arduino mega, la memoria interna de 2Kb que almacena datos sobre los horarios ingresados.

Para el manejo de corriente externa necesaria en el control de riego del encendido de una electroválvula el autor utilizó un **módulo relé**, por lo general este dispositivo no tiene gran diferencias entre uno y otro, todos cumplen con su principal característica, por lo que se concentró en los acoples del Arduino con la electroválvula, entradas y salidas de voltajes, se analizó el módulo relé dual que soporta hasta 110v para la conexión de la electroválvula.

El **teclado matricial**, como interface entre el usuario y el control, en el ingreso de nuevos valores al riego, sin que el usuario tenga la necesidad de modificar el código del control de riego implementado a la placa, en la búsqueda de este módulo el autor optó por un teclado numérico 4x3, dado que cumplía con los requisitos y funciones específicas necesarias por el control de riego.

Cuadro 3. 8 Programación de dispositivos

| Programación de Dispositivos | | | | |
|--|---|--|--|---|
| Actividad | Responsable Participantes | Insumo | Técnicas/ Herramientas | Productos |
| Desarrollo del código fuente del dispositivo | Responsables: El autor. Participantes: El autor. | Librerías de funcionamiento del dispositivos | Programación en lenguaje Arduino, software Arduino | Código fuente del dispositivo ver Anexo (III) |
| Simulación | Responsables: El autor. Participantes: El autor. | Código fuente del dispositivo | Mejoras del código en caso de que no cumpla con los requerimientos | Comprobación de la funcionalidad del código fuente Ver figura 4.6 |

-PROGRAMACIÓN DE DISPOSITIVOS

La programación de los dispositivos es uno de los procesos más importantes, ya que en este se desarrolló el código de programación, el autor realizó los respectivos análisis y pruebas del código que se implementó a la placa Arduino, realizó ensayos del código por separado de cada módulo, comprobando la buena interacción con la placa Arduino mega.

PROCESO DE CODIFICACIÓN

El proceso de codificación se lo realizó mediante la recolección de información ya analizada y procesada en los métodos y técnicas usadas, las cuales se encuentran representadas en el diagrama de flujo, esta información facilitó la dirección que tomó el código al momento de desarrollar el lenguaje de programación que se implementó en la placa Arduino mega junto con los actuadores a integrar en la automatización del sistema de riego logrando desarrollar la placa impresa y los circuitos integrados, descrito en el cuadro 3.9.

Cuadro 3. 9 Desarrollo de circuitos integrados

| Desarrollo de circuitos impreso | | | | | |
|---|---|----------------------|--|------------------------------|--|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/Herramientas | Productos |
| Desarrollo del diseño de HDL.(Desarrollo de Hardware libre) | Responsables : El autor. Participantes: El autor. | Esquema de impresión | Impresión en baquelita del circuito impreso | Programación HDL. | Placa Impresa Ver foto 4.2 |
| Verificación Lógica | Responsables : El autor. Participantes: El autor. | Placa impresa. | Cara frontal del diseño del circuito integrado | | Pruebas para detectar fallas en la placa |

- INTEGRACIÓN

El autor del desarrollo de control de riego Implementó e integró los módulos, que actuarían junto a los condensadores, resistencias, convertidores de voltajes, relé, todos estos necesarios en el proceso de ensamble del control de riego en la parte de integración del hardware en la placa impresa. Se realizaron pruebas pertinentes y necesarias para la verificación del correcto funcionamiento entre los actuadores, placa Arduino mega y placa impresa desarrollada por el autor, además de ensayos de funcionamiento en el comportamiento del HDL comprobando su buen funcionamiento junto a los componentes del desarrollo del control de riego se muestran en detalle en los cuadros 3.10 – 3.12 y foto 3.1.

Cuadro 3. 10 Integración

| Integración | | | | |
|--|---|-------------------------------|--|---|
| Actividad | Responsable Participantes | Insumo | Técnicas/ Herramientas | Productos |
| Integración de los módulos, programación de dispositivos | Responsables: El autor. Participantes: El autor. | Placa con el circuito impreso | Programación en lenguaje Arduino, software Arduino | Placa con software y hardware Integrada Ver foto 4.3 |

Cuadro 3. 11 Integración del software en el hardware

| Integración del software en el hardware | | | | | |
|---|---|---------------------------------------|--|---------------------------|----------------------------------|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/ Herramientas | Productos |
| integración del software en el hardware | Responsables: El autor. Participantes: El autor. | Placa Arduino, módulos ,código fuente | El proceso de cargar del código fuente se lo realizo mediante una pc con el software Arduino | Integración | Placa Programada Ver foto 3.3 |

Cuadro 3. 12 Verificación y Simulación

| Verificación y Simulación | | | | | |
|--|---|-----------------------|--|---|---|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/ Herramientas | Productos |
| Verificación de las etapas luego de la integración | Responsables: El autor. Participantes: El autor. | Ficheros del proyecto | En función de los requerimientos del sistema | Verificación del software integrado al hardware | Simulación Ver foto 3.4 y figura 4.7 |

Verificación y simulación desarrollada en los procesos que se efectúan en la tecnificación del riego.



Foto 3. 1 Simulación efectuada en el control de riego

-PROCESO DE PRUEBA Y PRESENTACIÓN

Se verificó el correcto funcionamiento del control de riego, observando el adecuado desempeño con los diferentes componentes periféricos tales como el

Modulo Sd, Modulo Rtc, Modulo, Relé, pantalla Lcd, necesarios en el desarrollo del proyecto visto y detallado en el cuadro 3.13 y foto 3.2.

Cuadro 3. 13 Liberación del hardware y software

| Liberación | | | | | |
|---------------------------------|---|--------------------|---|---------------------------|---|
| Actividad | Responsable Participantes | Insumo | Observaciones | Técnicas/ Herramientas | Productos |
| Liberación del control de riego | Responsables: El autor. Participantes: El autor. | Hardware terminado | Búsqueda de lugar óptimo para su implementación | | Producto estable y terminado Ver foto 3.5 y 4.4. |

Presentación del área donde se efectúan los procesos de riego, lugar donde se realizaron pruebas que demostraron su buena ejecución en los procesos de regío en el campo real.



Foto 3. 2 Foto de la implementación del sitio

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

A través de la implementación y desarrollo del control del sistema de Riego en la Finca “La Lucia” basado en hardware libre que ofrece la placa Arduino mega, se realizó la concierne sucesión de procesos metodológicos que se emplearon en la realización de la automatización del sistema de riego, se desarrollaron los procesos necesarios para la programación y configuración de los componentes del sistema de riego.

Se elaboró una encuesta en la que el resultado que se obtuvo es que el 86% de los habitantes si desea invertir tiempo en mejorar su sistema de riego automatizándolo y el 14% no lo cree tan necesario que se invierta mucho tiempo en automatizar este sistema gracias a esta información se creó un diseño en el manejo de la distribución del control del agua mostrado en la figura 4.1.

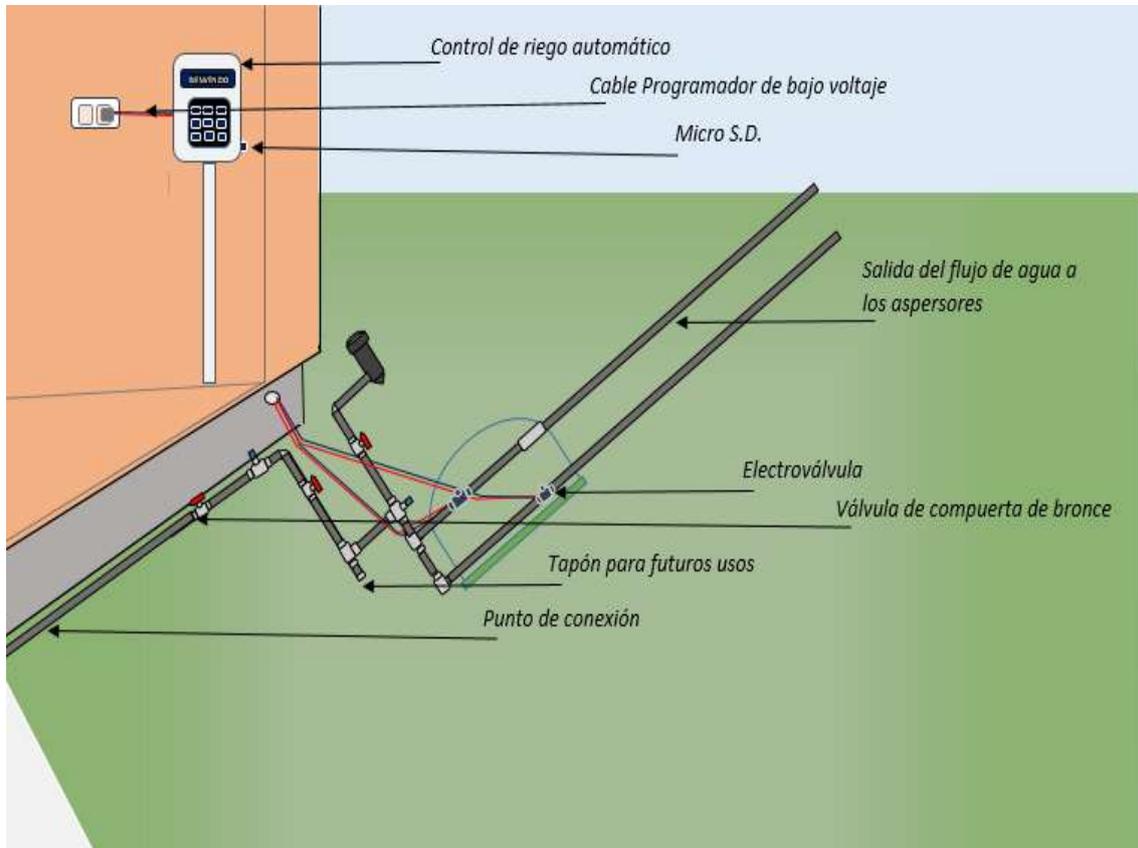


Figura 4. 1 Tecnificación del controlador de sistema de riego

Durante la investigación fue necesario realizar dos casos de uso para representar los requerimientos del sistema de riego, sustentar las pruebas de la implementación, verificar y validar cada elemento.

El primer caso de uso, evidenciado en la figura 4.2. hace referencia a las funciones del usuario, mismas que se detallan a continuación:

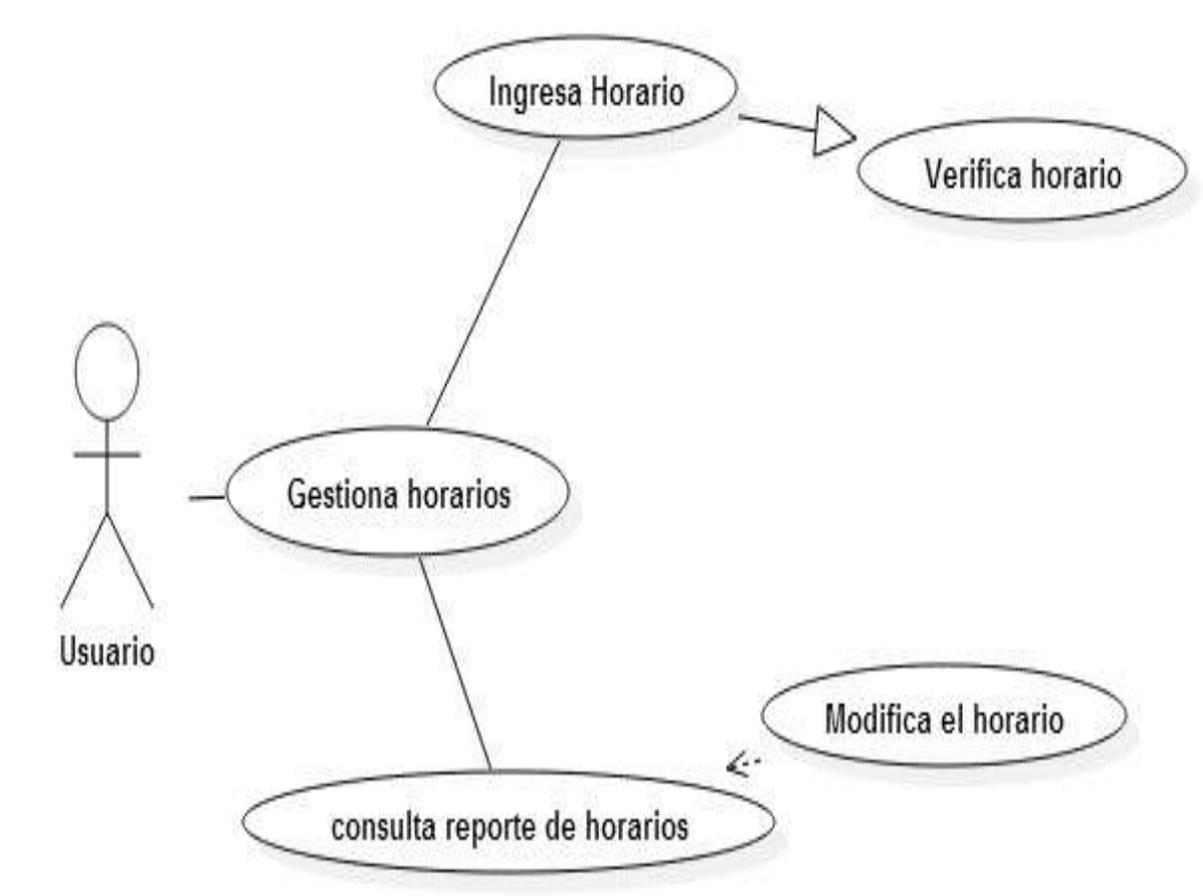


Figura 4. 2 Casos de uso (usuario del sistema)

El autor realizó un modelo Uml donde defino los requisitos, perspectivas y necesidades de la automatización, se inició con el desarrollo del diagrama de flujo de datos figura 4.3 obteniendo el diseño funcional del control de sistema de riego.

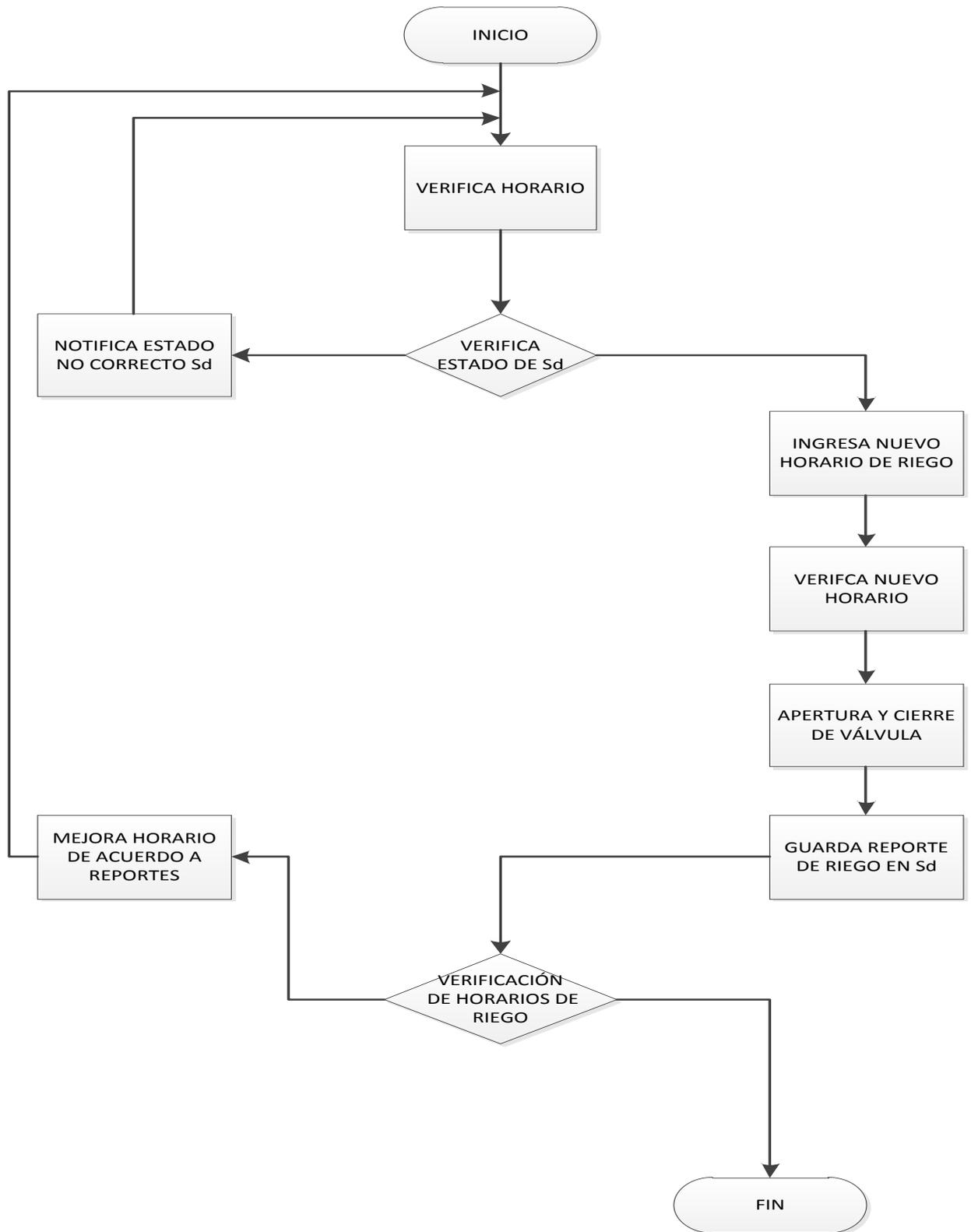


Figura 4. 3 Diagrama de flujo de los procesos de riego

Así mismo se realizó un diagrama de flujo del proceso de control de riego descrito en la figura 4.4.

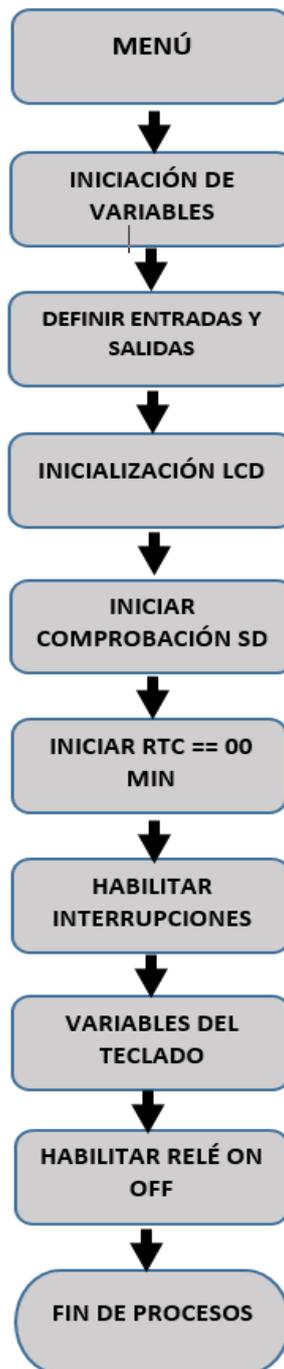


Figura 4. 4 Diagrama de flujo del proceso del control de riego

Para finalmente dar cumplimiento al primer objetivo específico el autor debió desarrollar los procesos de análisis, diseño y programación de códigos necesarios para la configuración del sistema de riego. A continuación se

muestra el esquema electrónico realizado en Fritzing, esquema electrónico junto a los componentes que conforman el control, diseño del circuito impreso diagrama electrónico elaborado para la implementación de los componentes de hardware en el control de riego junto a la placa Arduino, e integración del código fuente que se muestran las figuras 4.5 - 4.7.

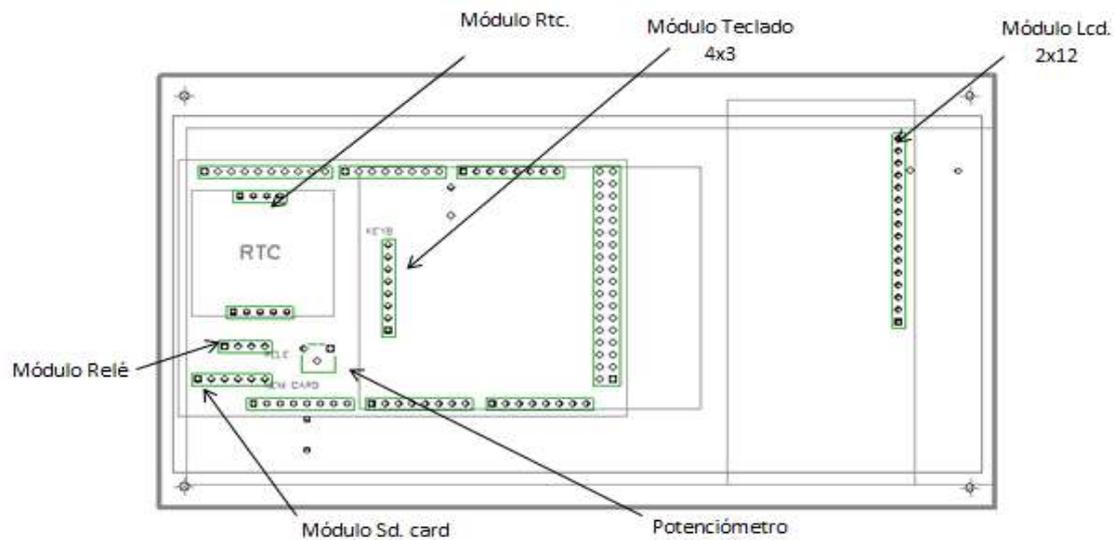


Figura 4. 5 Esquema de montaje electrónico

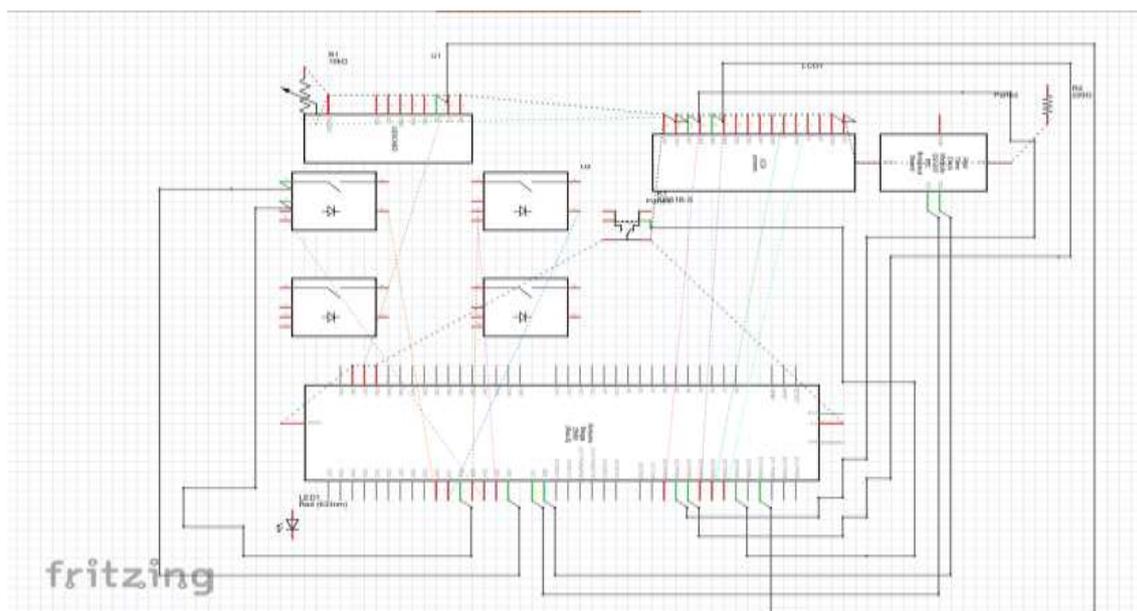


Figura 4. 6 esquema electrónico de los componentes que conforman el control de riego

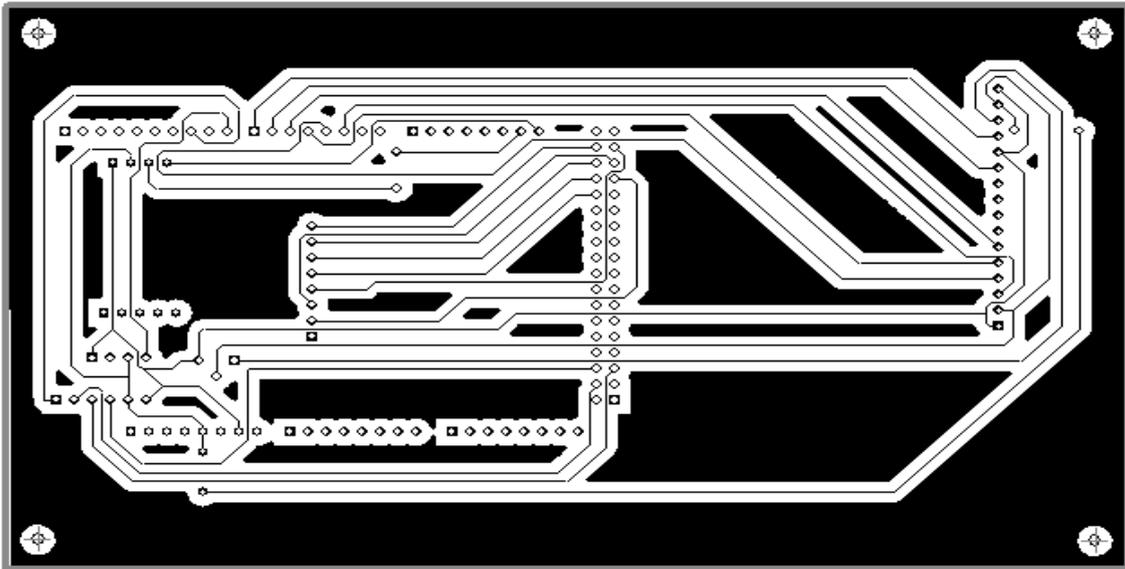


Figura 4. 7 Diseño del circuito impreso

Una vez concluidos los procesos anteriores se procedió a la programación del código que se implementó en la placa Arduino mega junto con los actuadores a integrar en la automatización del sistema de riego mostrados en la foto 4.1 y figura 4.8.

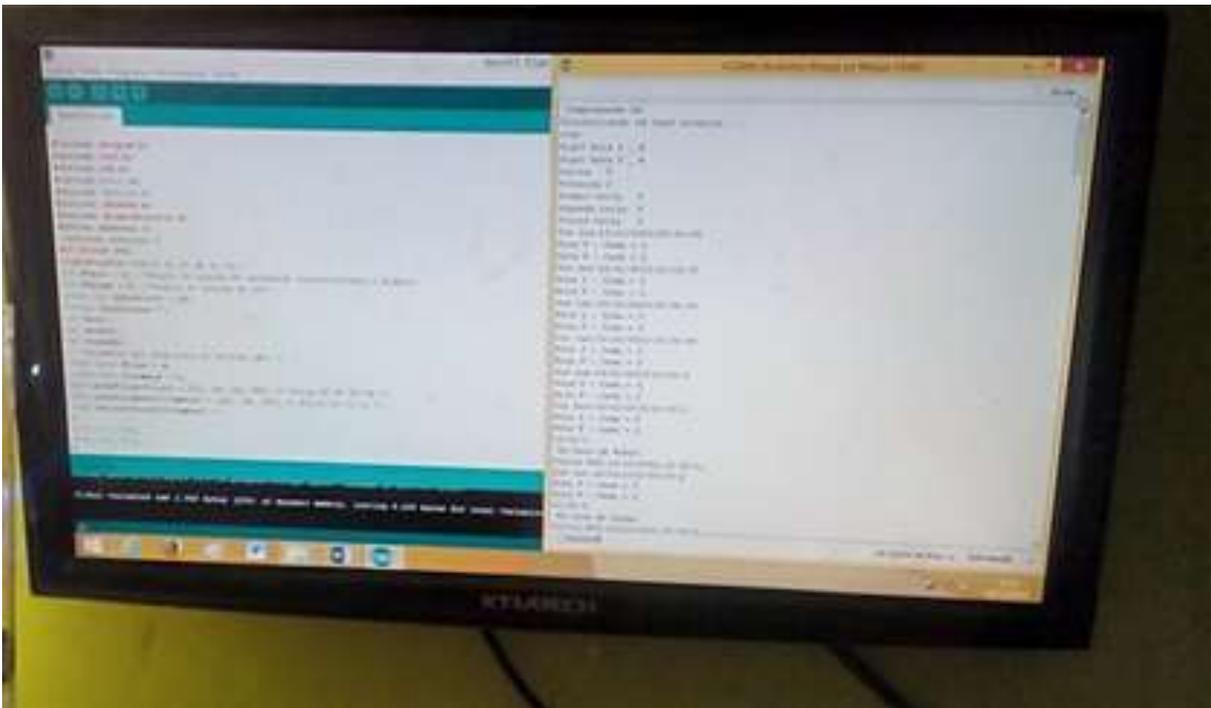
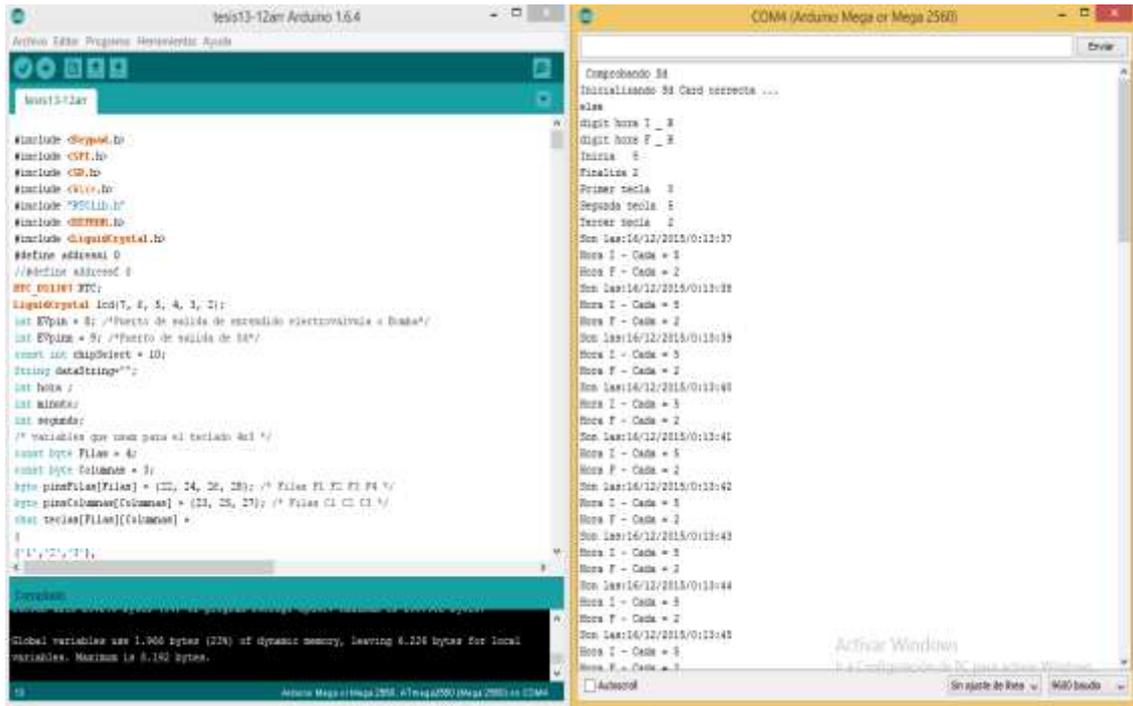


Foto 4. 1 Integración del código fuente al control de riego



The image shows two windows from an Arduino IDE. The left window, titled 'New13-12ar Arduino 1.6.4', displays the following code:

```

#include <Keyboard.h>
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <PS2Lib.h>
#include <EEPROM.h>
#include <LiquidCrystal.h>
#define address 0
//define address 0
//CUIDADO! NO:
LiquidCrystal lcd(7, 8, 9, 4, 1, 2);
int E1pin = 8; //Puerto de salida de escritura electrónica a Bauda?
int E2pin = 9; //Puerto de salida de Bauda?
const int chipSelect = 10;
String dataString="";
int letra ;
int minutos;
int segundos;
/* variables que usan para el teclado del */
const byte Filas = 4;
const byte Columnas = 3;
byte pinsFilas[Filas] = {23, 24, 26, 28}; // Filas P1 P2 P3 P4
byte pinsColumnas[Columnas] = {23, 25, 27}; // Filas C1 C2 C3
char teclado[Filas][Columnas] =
{
  {'1','2','3'},
  ...
}

```

The right window, titled 'COM1 (Arduino Mega or Mega 2560)', shows the serial monitor output:

```

Comprobando B4
Inicializando B4 Card correcta ...
else
digit Norte I _ N
digit Norte F _ N
Inicia B
Finaliza I
Primer tecla B
Segunda tecla B
Tercer tecla B
Sun Dec:16/12/2015/0:13:37
Hora I - Caida = 5
Hora F - Caida = 2
Sun Dec:16/12/2015/0:13:38
Hora I - Caida = 5
Hora F - Caida = 2
Sun Dec:16/12/2015/0:13:39
Hora I - Caida = 5
Hora F - Caida = 2
Sun Dec:16/12/2015/0:13:40
Hora I - Caida = 5
Hora F - Caida = 2
Sun Dec:16/12/2015/0:13:41
Hora I - Caida = 5
Hora F - Caida = 2
Sun Dec:16/12/2015/0:13:42
Hora I - Caida = 5
Hora F - Caida = 2
Sun Dec:16/12/2015/0:13:43
Hora I - Caida = 5
Hora F - Caida = 2
Sun Dec:16/12/2015/0:13:44
Hora I - Caida = 5
Hora F - Caida = 2
Sun Dec:16/12/2015/0:13:45
Hora I - Caida = 5
Hora F - Caida = 2

```

Figura 4. 8 Comprobación de la funcionalidad del código

El segundo objetivo de la investigación hace referencia a los prototipos de pruebas con la tarjeta Arduino. Respecto a esto la foto 4.2 Y 4.3 muestran la Baquelita con el circuito impreso perforado probada y verificada, lista para la integración con los módulos y placa Arduino y la Integración de los módulos a placa impresa.

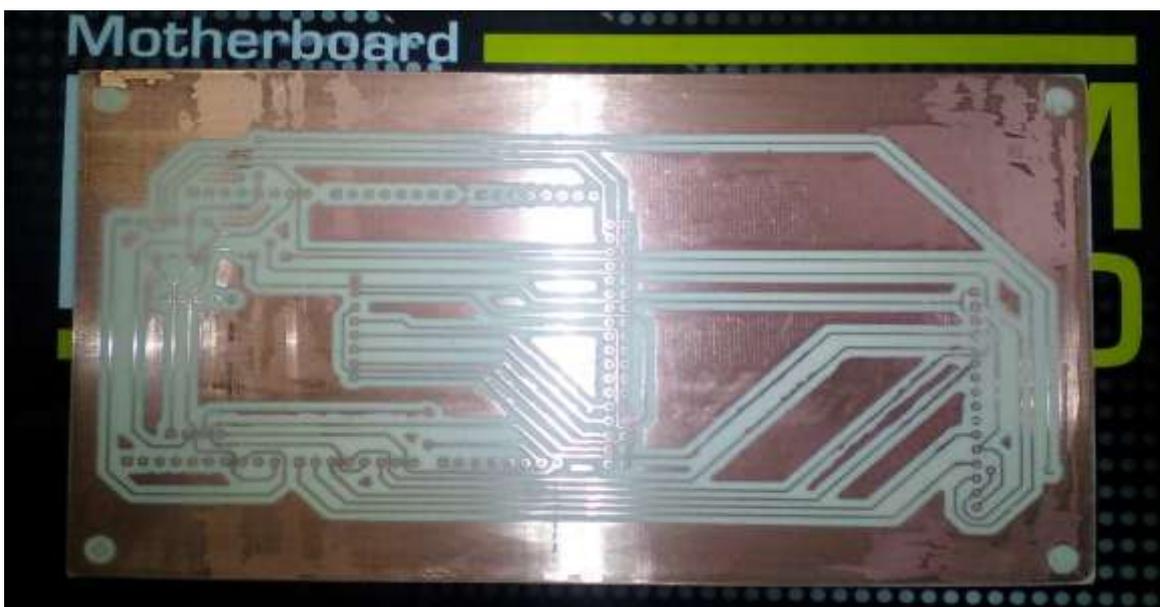


Foto 4. 2 Baquelita circuito impreso



Foto 4. 3 Integración de los módulos a placa impresa

Finalmente, se realizó la Implementación del sistema de riego en el área que se cultiva en la Finca “La Lucía”, resultando exitosa puesto que se efectuaron pruebas en prototipos, donde el autor verificó que se cumplieran cada uno de los requerimientos y necesidades que conllevaron a la implementación de la automatización al sistema de riego, una vez efectuado la automatización al sistema de riego en la Finca “La Lucia ” se evidenció el correcto funcionamiento de los equipos mediante pruebas y simulacros, donde se logró accezar a la información generada sobre los procesos efectuados en el riego de una manera ágil y eficaz. Verificación de reportes generados por el control de riego sobre los procesos que se efectúan en el riego, guardados en el módulo micro Sd instalada en el control de regío, en la figura 4.9 se muestra los resultados que genera el control.

| | | | | |
|------------|------------|----------|---------|----------|
| Inicio RSD | 16/08/2015 | 22:39:02 | Fin RSD | 22:40:00 |
| Inicio RSD | 16/08/2015 | 22:45:02 | Fin RSD | 22:46:00 |
| Inicio RSD | 16/08/2015 | 22:48:01 | Fin RSD | 22:51:00 |
| Inicio RSD | 16/08/2015 | 23:00:01 | Fin RSD | 23:03:00 |
| Inicio RSD | 16/08/2015 | 23:12:02 | Fin RSD | 23:15:00 |
| Inicio RSD | 16/08/2015 | 23:42:02 | Fin RSD | 23:44:00 |
| Inicio RSD | 16/08/2015 | 23:45:02 | Fin RSD | 23:47:00 |
| Inicio RSD | 16/08/2015 | 23:48:02 | Fin RSD | 23:50:00 |
| Inicio RSD | 16/08/2015 | 23:51:02 | Fin RSD | 23:53:00 |
| Inicio RSD | 16/08/2015 | 23:54:02 | Fin RSD | 23:56:00 |
| Inicio RSD | 16/08/2015 | 23:57:02 | Fin RSD | 23:59:00 |
| Inicio RSD | 17/08/2015 | 0:03:01 | Fin RSD | 0:05:00 |
| Inicio RSD | 17/08/2015 | 0:06:01 | Fin RSD | 0:08:00 |
| Inicio RSD | 17/08/2015 | 0:09:01 | Fin RSD | 0:11:00 |
| Inicio RSD | 17/08/2015 | 0:12:01 | Fin RSD | 0:14:00 |
| Inicio RSD | 17/08/2015 | 0:15:02 | Fin RSD | 0:17:00 |
| Inicio RSD | 17/08/2015 | 0:18:02 | Fin RSD | 0:20:00 |
| Inicio RSD | 17/08/2015 | 9:14:02 | Fin RSD | 9:15:00 |
| Inicio RSD | 17/08/2015 | 9:16:02 | Fin RSD | 9:17:00 |

Figura 4. 9 Simulación de reporte de procesos efectuados del control de riego

Índice de gasto de agua en el sistema de riego antes y después de la automatización implementada, por el autor del control de riego. Demostrando la efectividad en el control adecuado del agua en los riegos de la finca "La Lucía", el que se muestra detalladamente en el cuadro 4.1 y figura 4.10.

Cuadro 4. 1 Índice de gasto del agua

| Uso del agua sin control | | | |
|---------------------------------|--------------|-------------------------|-------------------|
| diario | Horas | Cantidad de agua | Total agua |
| 1 a 12 | 3 | 800 litros x h | 2400 litros |
| 13 a 24 | 4 | 800 litros x h | 3200 litros |
| | | | |
| Uso del agua con control | | | |
| diario | Horas | Cantidad de agua | Total agua |
| 1 a 12 | 2 | 800 litros x h | 1600 litros |
| 13 a 24 | 2 | 800 litros x h | 1600 litros |

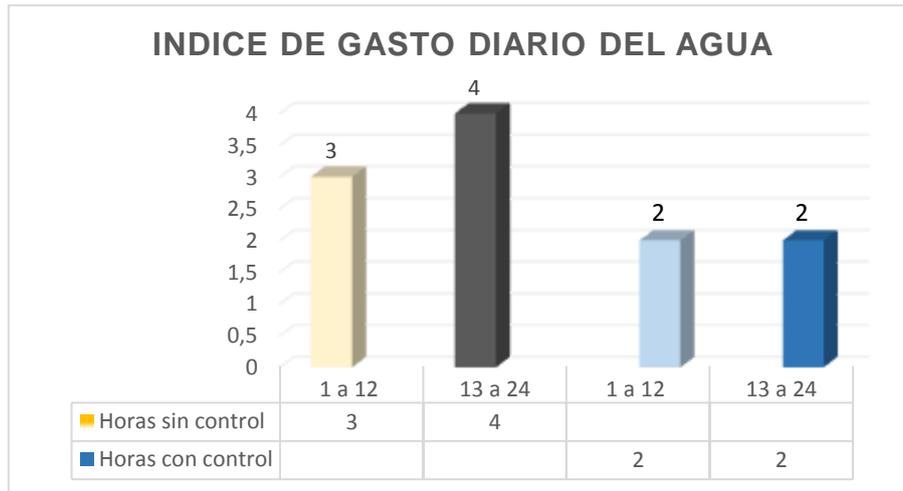


Figura 4. 10 Índice de horas del gasto de agua en el riego

De acuerdo al trabajo desarrollado por Matheus (2011) la necesidad de implementar sistema de riego a los cultivos para mejorar el uso del agua es de vital importancia, para el efecto el mencionado autor implementó un sistema de riego para uso del cultivo de papa, teniendo como objetivo desarrollar un sistema de riego por aspersion basado en el manejo manual en sectores determinados, su única limitante era que no podría controlar el sistema de riego de forma automática y llevar el registro de los procesos de riego de acuerdo al cultivo. Con la implementación de un control elaborado con una placa Arduino y componentes, que se muestra en la foto 4.4 el usuario logró acceder al control y programar la hora en que se necesite ser ejecutado el riego, logrando tener un acceso a los reportes que genera este sistema para mejoras al control del tiempo de riego en base a los resultados obtenidos.

El usuario logro acceder al control de riego en horas adecuadas gracias a la implementación del hardware programable en el sistema de regío, personalizando la irrigación del sembrío de acuerdo a cada tipo de cultivo, obteniendo una mayor eficacia en la aspersion y aprovechamiento del agua .



Foto 4. 4 Hardware instalado y probado

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

El autor del trabajo de investigación se permite concluir que:

- ✓ El desarrollo de una ficha técnica en base al levantamiento de información del proyecto desarrollado logra automatizar el sistema de riego, solventando las distintas necesidades y requerimientos del mismo.
- ✓ En la realización de los prototipos de prueba confirma el buen uso del control de riego en el área de sembrío, lo que influye de manera positiva en la eficiencia y productividad de los cultivos, terrenos y sembríos aledaños.
- ✓ Una vez se implementó y probó la automatización en el control de riego se confirmó la eficiente distribución del agua destinada al riego de los sembríos en horas adecuadas y oportunas de acuerdo a cada tipo.

5.2 RECOMENDACIONES

Finalizado el proyecto sobre la automatización del control de riego con placa Arduino para la finca “La lucia” el autor se permite recomendar que:

- ✓ Se debe elaborar una buena ficha técnica, investigar y adquirir mayor conocimiento sobre la creación de fichas técnicas en el proceso de realización de proyectos ya sean estos de tipo software o hardware.
- ✓ Se debe elaborar un plano de la interfaz del sistema de riego, de manera comprensible para el usuario, ya que en este se detallan los sitios donde serán instalados los equipos para aportar a la eficiencia y productividad del riego a los cultivos.
- ✓ Se debe utilizar los reportes generados como bitácora por el control de riego para cumplir con los requerimientos y necesidades del usuario, en la distribución eficiente del agua, en los sembríos a horas adecuadas.

BIBLIOGRAFÍA

Agudelo, R. 2005. Automatización de sistema de riego. Pontificia Universidad Javeriana.

Álvarez, C; Soto, A; Watkins, F; 2009. Simulación de Controles Digitales. Chile, CL. Revista Chilena de Ingeniería. vol. 17. N. 3. p 309-316.

Álvarez, J; 2013. Metodología para Desarrollo de Software Libre. Fundación CENDITEL. Versión 2.

Ambientum.com. 2008. Irrigación automatizada en áreas de agrícolas. ARDUINO - Homepage (última modificación Julio 2011). Formato PDF. Disponible en: <http://www.Arduino.cc>

Libro Arduino. 2012. Equipo de cosas mecatrónicas "Electrónica definición. 1 ed. p 17.

ARDUINO. 2011. Básicos-Arduino. Formato PDF. Disponible en: <http://www.Arduino.cc/es>

Benet C. 2005. Definición de Software. Libro Ingeniería del Software. p 13.

Díaz, A. 2012. Hardware definición publicado en Universidad Autónoma del Estado de Hidalgo Pdf. P. 2-4.

Méndez, N. y Gómez, J. 2010. Electrónica Moderna Definición de Prototipos electrónicos CEKIT.

ESPOL (Escuela Superior Politécnica de Manabí). 2006. Efectos de riego .Revista Tecnológica ESPOL. EC. Vol. 19. N. 1. p 25-32.

Evans, B. 2007. Arduino Programming Notebook. Disponible en http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf

García I. y Briones, G. 2009. Sistemas de riego por Aspersión y Goteo. Editorial Trillas. 2 ed. p 12-44.

- Galavis, S. y Álvarez, G. 2010. La encuesta de opinión estudiantil: un sistema de información para la evaluación por competencias de la actividad docente. Revista de la Facultad de Ingeniería Universidad Central de Venezuela. Caracas. Vol. 25. n. 3.
- Herrador E. 2009 “hardware ampliable de código abierto” Guía de Usuario de Arduino.
- Hunter Industries Incorporated 1940. Diamond Street, San Marcos, California 92078. P 3. Disponible en: www.hunterindustries.com/global.
- Pomares, J. 2009. Lenguaje Arduino, Manual para Arduino Universidad de Alicante Grupo de Innovación Educativa en Automática.
- Lugo, O; Quevedo, A; Bauer, J; Del valle, D; Palacios, E; Águila, M; 2011. “Prototipo para automatizar Un sistema de Riego Multicultivo” Revista Mexicana De Ciencias agrícolas. Vol. 2. N. 5. p 659-672.
- Matheus, F. 2011. Diseño de un sistema de riego por aspersión para el cultivo de la papa Universidad de Los Andes. Venezuela. P 77.
- Medrano, A. 2011. Metodología de Desarrollo en Hardware Libre. (En línea). Consultado, 14 de mayo del 2012. Formato PDF. Disponible en <http://hl.cenditel.gob.ve/intro/metodologia/>
- Miguel, A; de Pablo Hdez. Y C. de Pablo S. Revista de Ciencia, Tecnología y Medio Ambiente volumen vii. Año 2010 Pg. 4.
- Monk S, 2012. 30 proyectos con Arduino publicado Editorial Estribor .S.I. formato PDF. 2 ed. p 61-72
- Moreno, M. 1987. Introducción a la Metodología de la Investigación Educativa. Editorial Progreso. 1 ed. p 41 – 42.
- Torrente, O. 2013. Libro de curso práctico de formación en Arduino, Señales analógicas y señales digitales. p 7-8.
- Toapanta, D. 2012. Tesis previa la obtención de título de Ingeniero Electrónico” Diseño e implementación de un módulo didáctico para realizar el proceso

de verificación y escariado de piezas mecánicas” disponible en <http://repositorio.espe.edu.ec/xmlui/handle/21000/2715/browse?value=toapanta+iza%2c+diego+pa%ef%bf%bd&type=author>

Páez, M. y Gerard. 2007. Creación de un ambiente Tecnológico para el Diseño de Circuitos Integrados. CEMISID – Facultad de Ingeniería. Universidad de Los Andes.

Pallas, A. 2007. “Micro controladores: fundamentos y aplicaciones con PIC” Marcombo.

Ponsa, P. y Granollers, T. 2011. Diseño y Automatización industrial: definición Automatización. p 2

Silva, U. 2010. Los residuos electrónicos: Un desafío para la sociedad del conocimiento en América Latina y el Caribe. Montevideo, UY. Plataforma UNESCO. p 188.

____.S.F. Datasheet. (En línea). Consultado 13 de Agosto. 2015. Formato PDF. Disponible en http://www.datasheetcatalog.org/datasheets2/16/169021_1.pdf

Vega E, Santamaría P, Rivas T. 2014. Internet de los objetos empleando Arduino para la gestión eléctrica domiciliaria Disponible en:http://www.scielo.org.co/scielo.php?pid=S012081602014000200001&script=sci_arttext

Vinueza P. 2015. Diseño y construcción de brazo robótico utilizando servo motores para el campo de la agricultura Disponible en <https://secure.orkund.com/view/document/13395111-704802-956902/download>

ANEXOS

ANEXO 1: PRESUPUESTO Y FUENTE DE FINANCIAMIENTO

| Actividad | Detalle | Cantidad | Valor unitario | Valor total | Fuente de financiamiento |
|--|---------------------|----------|----------------|-------------|--------------------------|
| Entrevista | Pasajes | 4 | 1 | 4 | El Autor |
| | Lápiz | 2 | 0,5 | 1 | El Autor |
| | Hojas | 10 | 0,01 | 0,1 | El Autor |
| Comprender problemática | Hojas | 1 | 5 | 5 | El Autor |
| | Internet | 15 | 1 | 15 | El Autor |
| Analizar los campos requeridos para cada proceso | Hojas | 10 | 0,01 | 0,1 | El Autor |
| | Lápiz | 2 | 0,5 | 1 | El Autor |
| Diseñar mapa de implementación | Lápiz | 2 | 0,5 | 1 | El Autor |
| | Cartulina | 2 | 1 | 2 | El Autor |
| | Borrador | 2 | 0,5 | 1 | El Autor |
| Desarrollar los procesos de programación | Computadora | 1 | 860 | 860 | El Autor |
| | Internet | 100 | 1 | 100 | El Autor |
| | Open Source | 1 | 0 | 0 | El Autor |
| Realizar prototipos de pruebas | Protoboard | 1 | 25 | 25 | El Autor |
| | Pantalla lcd | 1 | 20 | 20 | El Autor |
| | Válvula electrónica | 1 | 70 | 70 | El Autor |
| | Cables de conexión | 1 | 10 | 10 | El Autor |
| | Diodos Led | 1 | 20 | 20 | El Autor |
| | resistencias | 10 | 1 | 10 | El Autor |
| | Lector sd | 2 | 20 | 40 | El Autor |
| | Actuadores | 1 | 160 | 160 | El Autor |
| | Memoria Sd | 2 | 10 | 20 | El Autor |
| | Placa Arduino | 2 | 40 | 80 | El Autor |
| Asesoría Externa | Ing. Luis Santana | 1 | 100 | 100 | |
| Implementar el sistema de riego | Talento Humano | 12 | 5 | 60 | |
| | pasajes | 12 | 1 | 12 | El Autor |
| | Herramientas | 1 | 40 | 40 | El Autor |
| Capacitar | Talento Humano | 1 | 0 | 0 | ESPAM MFL |
| Total | | | | 1557,2 | |

ANEXO 2: ENCUESTA REALIZADA A LOS HABITANTES DE LA FINCA LA**ENCUESTAS REALIZADAS A LOS HABITANTES DE LA FINCA****LA LUCIA**

1. ¿Usted cree que el sistema de riego utilizado en sus plantaciones es totalmente eficiente y confiable?

Si ()

No ()

Tal vez ()

2. ¿Usted cree que el sistema de riego se podría mejorar con su automatización?

Si ()

No ()

Definitivamente No ()

3. ¿Usted cree que se debe invertir tiempo en mejorar el sistema de riego automatizándolo?

Si ()

No ()

4. ¿Cree usted que mejorara el manejo del agua y se evitara el gasto innecesario con esta solución?

Si ()

No ()

No lo sé ()

5. ¿Estaría dispuesto a permitir automatizar el sistema de riego y en qué porcentaje?

25% ()

50% ()

100% ()

Otro ()

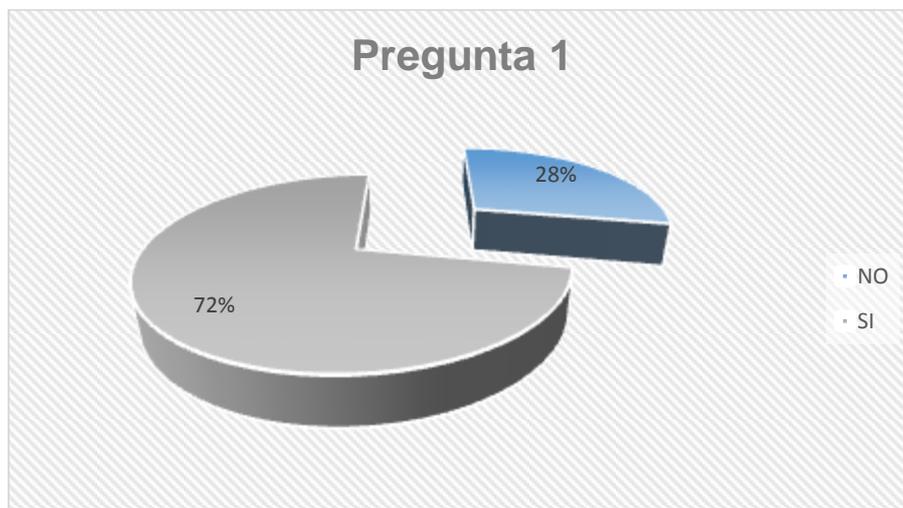
ENCUESTA PREGUNTA Y GRAFICO # 1

PREGUNTA # 1: Usted cree que el sistema de riego utilizado en las plantaciones es totalmente eficiente y confiable?

| ALTERNATIVAS | TOTAL | |
|--------------|-------|-----|
| | # | % |
| Si | 5 | 72 |
| No | 2 | 28 |
| TOTAL | 7 | 100 |

Tabla: Pregunta 1

Fuente: Habitantes de la Finca "La Lucia"



Los resultados que se obtuvieron señalaron que el 72% de los habitantes de la finca Lucia se encuentran descontento con el sistema de riego usado en los cultivos, mientras que el 28% están satisfecho con el sistema de riego usado.

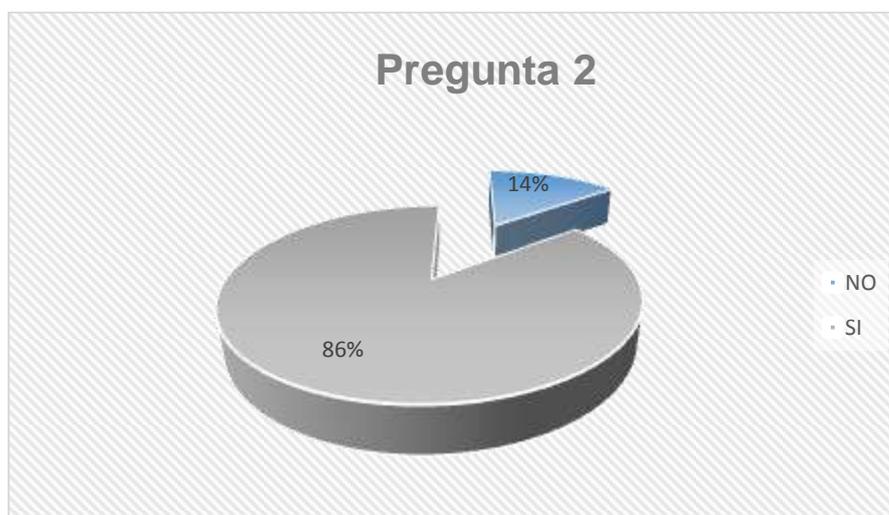
PREGUNTA Y GRAFICO # 2

PREGUNTA # 2: Usted cree que el sistema de riego se podría mejorar con la automatización?

| ALTERNATIVAS | TOTAL | |
|--------------|-------|-----|
| | # | % |
| Si | 6 | 83 |
| No | 1 | 17 |
| TOTAL | 7 | 100 |

Tabla: Pregunta 2

Fuente: Habitantes de la Finca “La Lucia”

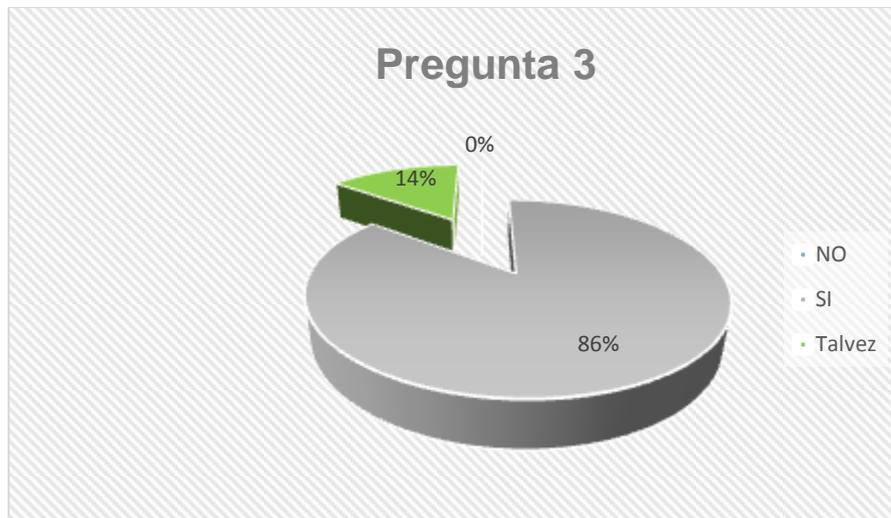


El resultado que se obtuvo en esta pregunta nos da a conocer la aceptación de automatizar el sistema de riego ya que el 83% está seguro de que la automatización es beneficiosa para los cultivos y el 17% no está de acuerdo en que sea una buena solución

PREGUNTA Y GRAFICO # 3

PREGUNTA # 3: Usted cree que se debe invertir tiempo en mejorando el sistema de riego automatizándolo?

| ALTERNATIVAS | TOTAL | |
|--------------|-------|-----|
| | # | % |
| Si | 6 | 86 |
| No | 0 | 0 |
| Tal vez | 1 | 14 |
| TOTAL | 7 | 100 |

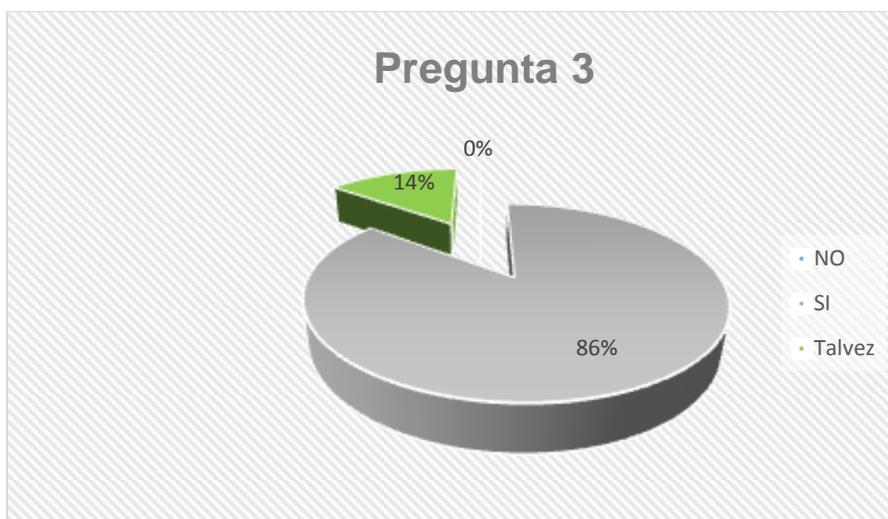


El resultado que se obtuvo en esta pregunta es que el 86% de los habitantes si desea invertir tiempo en mejor su sistema de riego automatizándolo y el 14% no lo cree tan necesario que se invierta mucho tiempo en automatizar este sistema.

PREGUNTA Y GRAFICO # 4

PREGUNTA # 4: Cree Usted que mejorará el manejo del agua y se evitara el gasto innecesario con esta solución?

| ALTERNATIVAS | TOTAL | |
|--------------|-------|-----|
| | # | % |
| Si | 6 | 86 |
| No | 0 | 0 |
| Tal vez | 1 | 14 |
| TOTAL | 7 | 100 |

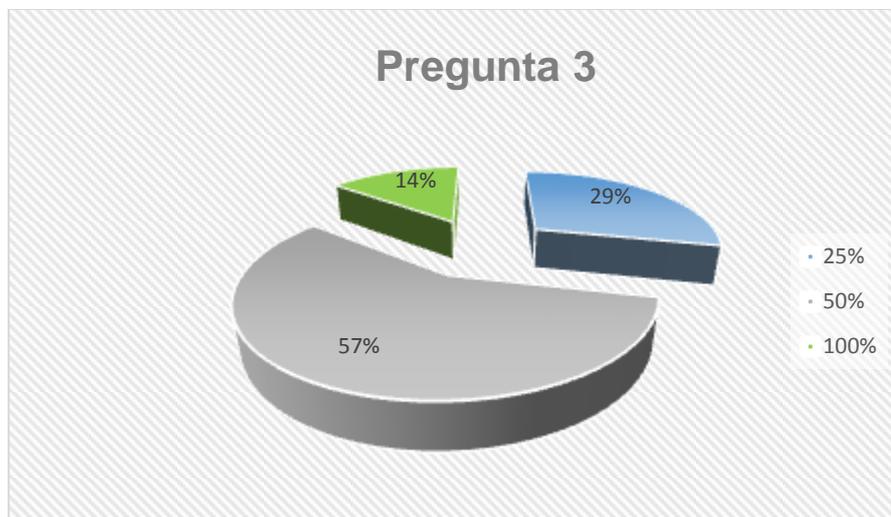


El resultado que se obtuvo en esta pregunta es que el 86% de los habitantes si desea invertir tiempo en mejor su sistema de riego automatizándolo y el 14% no lo cree tan necesario que se invierta mucho tiempo en automatizar este sistema.

PREGUNTA Y GRAFICO # 5

PREGUNTA # 4: Estaría dispuesto a permitir automatizar el sistema de riego y en qué porcentaje?

| ALTERNATIVAS | TOTAL | |
|--------------|-------|-----|
| | # | % |
| 25% | 2 | 29 |
| 50% | 4 | 57 |
| 100% | 1 | 14 |
| TOTAL | 7 | 100 |



El resultado que se obtuvo es que el 86% de los habitantes si desea invertir tiempo en mejorar su sistema de riego automatizándolo y el 14% no lo cree tan necesario que se invierta mucho tiempo en automatizar este sistema.

ANEXO 3: FICHA TÉCNICA DE DESARROLLO DE HARDWARE Y SOFTWARE

FICHA TÉCNICA

ESPECIFICACIÓN DE REQUISITOS HARDWARE Y SOFTWARE DEL CONTROL DE RIEGO

INTRODUCCIÓN

Automatización del sistema de riego con una placa Arduino para la finca “la lucia”

La Automatización del sistema de riego realizará los procesos de control de regío, dará un informe detallado acerca de las horas en que se realizó el riego en las áreas de sembríos, controlara el flujo de agua en los aspersores en las horas que crea conveniente el usuario, personalizando el regío por cada tipo de cosecha.

1. PROPÓSITO

Controlar los procesos del sistema de Riego automatizándolos

Mejorar el rendimiento en la producción agrícola

Obtener reporte detallado de las horas en que se realizó cada riego

1.2 Alcance

En el alcance intervino los Habitantes de La finca “La Lucia”, y el autor

Controlar el riego en las áreas de sembríos

Diseño de los requisitos claros del sistema de riego en un diagrama de caso de uso figura 4.2.

Diseño de diagrama de flujo de procesos de riego que ejecutara el sistema de riego figura 43.

2. REQUERIMIENTOS DE HARDWARE Y SOFTWARE

2.1 RESUMEN DE REQUERIMIENTOS HARDWARE

Esta sección entrega un resumen de todos los requerimientos hardware.

RH1 Tecnificar los procesos de riego

RH2 Controlar el flujo de agua a horas adecuadas

RH3 Accesibilidad al usuario mediante un teclado para la modificación de valores en el horario de riego

2.2S RESUMEN DE REQUERIMIENTOS SOFTWARE

Esta sección entrega un resumen de todos los requerimientos software.

RH1 Controlar y mantener el software actualizado en cuanto a la hora, fecha

RH2 Visualización de los procesos que realiza el código en el control

RH3 desarrollar y almacenar un reporte detallado en la memoria Micro Sd.

3. DETALLE DE REQUERIMIENTOS HARDWARE Y SOFTWARE

3.1 Identificación de requisitos funcionales

Función del control

- Controlar el sistema de riego
- Generar y almacenar informes del riego

Entradas

- Datos en relación al horario
- Datos programados del riego

Salidas

- Reporte de riego
- Reporte de la Hora del riego
- Riego generado
- Visualización de horarios
- Visualización de estado del riego
- Visualización de comprobación de datos en SD

3.2 Especificaciones eléctricas y funcionales

3.2.1 RH1 – Implementación de una placa Arduino, y actuadores para administrar el control de riego

RH1.1 Implementación de un Módulo Rtc. (reloj)

RH1.2 implementación de una electroválvula en el control de riego de 110v

3.3 Requerimientos mecánicos

Los componentes o módulos implementados a la placa Arduino como lo es Lcd, lector Sd, Relé, Rtc, se incluyeron en la placa del circuito impreso de acuerdo a necesidades y requerimientos que el control de riego necesite o establezca el usuario

3.3.1 Requerimientos del desarrollo de lenguaje del software

RH2 – Lenguaje Arduino

RH2.1 Software Arduino 1.6.4, Librerías, Sistema Operativo Windows, Fritzing.

RH2.2 Acceso a interface del software Arduino compatible con módulos de hardware.

3.3.1 Requerimientos del Circuito Impreso PCB

RH2 – Esquema del diseño impreso en PCB

RH2.1 Implementación del diseño del circuito impreso

RH2.2 Implementación de pistas de alimentación de 0.150 pulgadas de ancho para poder soportar una corriente de 5v

RH3 – Dimensión, forma, y orificios de montaje

RH3.1 Implementación de borneras en el borde inferior (ver diagrama adjunto)

RH3.2 Dimensiones tarjeta de acuerdo al diagrama adjunto

3.3.2 Requerimientos de Control, Indicadores y Conectores

RH4 – Switch Principal

RH4.1 botón de encendido de resistencia de hasta 32v

3.3.3 Ubicación de Componentes

RH5 – Ubicación de módulos Micro Sd., Relé

RH5.1 Ubicación de un módulo lector de Sd en el cuadrante izquierdo, lejos de la entrada de corriente

RH5.2 Ubicación de un módulo Relé en el cuadrante inferior derecho lejos de los micros controladores

3.4 Restricciones Específicas

C1 Manipulación del dispositivo con distintos cambios de voltajes

3S REQUISITOS ESPECÍFICOS

Requisitos comunes de los interfaces

Interfaces de usuario

El interface del producto consta de una pantalla Lcd. De 2x12 por la cual se visualizara los procesos que ejecuta el control de riego en tiempo real junto con la programación integrada en el dispositivo hardware,

ANEXO 4: CÓDIGO FUENTE ARDUINO IMPLEMENTADO EN EL CONTROL DE RIEGO

Código Fuente del control de riego

```

#include <Keypad.h>
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include "RTClib.h"
#include <EEPROM.h>
#include <LiquidCrystal.h>
#define addressi 0
//#define addressf 0
RTC_DS1307 RTC;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
int EVpin = 8; /*Puerto de salida de encendido electroválvula o Bomba*/
int EVpinn = 9; /*Puerto de salida de Sd*/
const int chipSelect = 10;
String dataString="";
int hora ;
int minuto;
int segundo;
/* variables que usan para el teclado 4x3 */
const byte Filas = 4;
const byte Columnas = 3;
byte pinsFilas[Filas] = {22, 24, 26, 28}; /* Filas F1 F2 F3 F4 */
byte pinsColumnas[Columnas] = {23, 25, 27}; /* Filas C1 C2 C3 */
char teclas[Filas][Columnas] =
{
{'1','2','3'},
{'4','5','6'},
{'7','8','9'},
{'*','0','#'}
};
Keypad teclado = Keypad (makeKeymap(teclas),
pinsFilas,pinsColumnas,Filas,Columnas);
char teclal;
char teclaF;
int numero;
/* codigo contador */
int contador =0;
/*----- Codigo cambio Horario-----*/
int value;
int value2;
int valuef;

```

```

    int numeroexacto1 ;
    int numeroexacto2;
    int numeroexacto3;
    int inicioh;

void setup ()
{
  /* Inicializamos el puerto serie, wire y el modulo RTC */
  pinMode (EVpin, OUTPUT);
  Serial.begin(9600);
  Wire.begin();
  RTC.begin();
  /* Si quitamos el comentario de la linea siguiente, se ajusta la hora y la fecha
  con la del ordenador */
  //RTC.adjust(DateTime(__DATE__, __TIME__));
  lcd.begin(16,2);

  lcd.setCursor(3,0);
  lcd.print("BIENVENIDOS");

  byte iconocomp[8]={
    B00000,
    B01010,
    B01010,
    B01010,
    B00000,
    B10001,
    B01110,
    B00000,
  };
  lcd.createChar(5,iconocomp);
  lcd.setCursor(8,1);
  lcd.write(5);
  delay(1000);
  lcd.clear();
  lcd.setCursor(7,0);
  lcd.print("A");
  lcd.setCursor(0,1);
  lcd.print("Sistema de Riego");
  delay(1000);
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("Automatizado");
  lcd.setCursor(2,1);
  lcd.print("De la Finca");
  delay(2000);
  lcd.clear();
  lcd.setCursor(4,0);

```

```

lcd.print("La Lucia");
lcd.setCursor(8,1);
lcd.print("W.R.C.R.");
// delay(1000);
// lcd.clear();

byte registrada[8]={
  B00001,
  B00110,
  B01010,
  B10010,
  B10010,
  B01010,
  B00110,
  B00001,
};
lcd.createChar(5,registrada);
lcd.setCursor(5,1);
lcd.write(5);
byte Marca[8]={
  B10000,
  B10100,
  B01010,
  B01001,
  B10001,
  B01010,
  B00100,
  B10000,
};
lcd.createChar(6,Marca);
lcd.setCursor(6,1);
lcd.write(6);
delay(2000);
lcd.clear();

lcd.setCursor(0,0);
lcd.print("La Hora Programada");
lcd.setCursor(0,1);
lcd.print("En el riego");
delay(2000);
lcd.clear();

/*se ingresara en las variables las horas programadas con el teclado */

lcd.setCursor(0,0);
value= EEPROM.read(addressi);
// value= EEPROM.read(addressi+1);

lcd.print(String("Inicia Cada ")+value+"(H)");

```

```

lcd.setCursor(0,1);
valuef= EEPROM.read(addressi+2);
lcd.print(String("Finaliza Cada")+valuef+"H");
delay(2500);
lcd.clear();

```

```
//Fin Ingreso Horas
```

```

lcd.setCursor(0,1);
lcd.print("Inicializando.");
delay(500);
lcd.clear();
lcd.setCursor(0,1);
lcd.print("Espere.");
delay(500);
lcd.clear();
lcd.setCursor(0,1);
lcd.print("Inicializando..");
delay(500);
lcd.clear();
lcd.setCursor(0,1);
lcd.print("Espere...");
delay(500);
lcd.clear();
lcd.setCursor(0,1);
lcd.print("Inicializando...");
delay(500);
lcd.clear();

```

```

{
  /* -----Codigo parta comprobar el estado de la memoria SD----- */
  Serial.println(" Comprobando Sd");
  pinMode(chipSelect,OUTPUT);
  lcd.print(" Comprobando SD");
  delay(500);
  lcd.clear();
  if(!SD.begin(chipSelect))
  {
    Serial.println("Fallo de inicializando Sd Card ... ");// verificando la Sd Card
    lcd.print("Fallo Sd Card ... ");
    delay(2000);
    lcd.clear();
  }
  Serial.println("Inicializando Sd Card correcta ... "); // verificando la Sd Card
  lcd.print("Sd Card correcta ... ");
  delay(2000);
  lcd.clear();
}

```

```

}
}

void loop ()
{
/* variables para el ingresos a la memoria EEPROM*/

// int numeroexacto1 ;
// int numeroexacto2;
// int numeroexacto3;

int variable1;
int variable2;
int variable3;

{

/* validacion contador */

for (int c=0; c<1; c++)
{
int f= contador ++;

if (f == 0)
{

/* configuracion para mostrar la hora */
Serial.println("else");
DateTime now = RTC.now();
hora =(now.hour());
lcd.setCursor(1,1);
lcd.print(hora);
lcd.clear();

lcd.setCursor(0,0);
Serial.println("digit hora I _ R ");
lcd.print("Digit hora I _ R");
lcd.setCursor(0,1);
Serial.println("digit hora F _ R");
lcd.print("Digit hora F _ R");
delay(2000);
numeroexacto2 = variable2 -48;

variable1= teclado.waitForKey();
numeroexacto1 = variable1 -48;
lcd.setCursor(13,0);
lcd.print(numeroexacto1);

```

```

if (numeroexacto1 == 0)

    {
        numeroexacto1 = 0 + numeroexacto1;
        variable2= teclado.waitForKey();
        numeroexacto2 = variable2 -48;
        lcd.setCursor(14,0);
        lcd.print(numeroexacto2);
    //    }
    //    if (numeroexacto2 == 1)
    //    {
    //        lcd.setCursor(14,0);
    //        lcd.print("_");
    //        variable2= teclado.waitForKey();
    //        numeroexacto2 = variable2 -48;
    //        lcd.setCursor(14,0);
    //        lcd.print(numeroexacto2);

    //    }
    else if (numeroexacto1 == 1)

    {
        numeroexacto1 = numeroexacto1 + 9;
        variable2= teclado.waitForKey();
        numeroexacto2 = variable2 -48;
        lcd.setCursor(14,0);
        lcd.print(numeroexacto2);

    //    }
    else if (numeroexacto1 == 2)

    {
        numeroexacto1 = numeroexacto1 + 18;
        variable2= teclado.waitForKey();
        numeroexacto2 = variable2 -48;
        lcd.setCursor(14,0);
        lcd.print(numeroexacto2);
    //    lcd.clear(int (numeroexacto2));

        Serial.print(numeroexacto2);
        Serial.print("l1");
        if (variable2 > 24)
            lcd.setCursor(14,0);
        lcd.print("_");
        variable2= teclado.waitForKey();

        Serial.print(numeroexacto2);
    }

```

```

Serial.println("I2");
numeroexacto2 = variable2 -48;
  lcd.setCursor(14,0);
lcd.print(numeroexacto2);

  inicioh = numeroexacto1+numeroexacto2;
  Serial.print(numeroexacto1);
  Serial.print(numeroexacto2);
  Serial.print(inicioh);

}

variable3= teclado.waitForKey();
numeroexacto3 = variable3 -48;
inicioh = numeroexacto1+numeroexacto2;
Serial.print("Inicia ");
Serial.println(inicioh);
Serial.print("Finaliza ");
Serial.println(numeroexacto3);
{
  if (numeroexacto3 < inicioh)
  {
    lcd.setCursor(17,0);
    lcd.print(variable3);

  }
  else
  {
    lcd.setCursor(0,0);
    lcd.print("VUELVA INGRESAR");
    lcd.setCursor(0,1);
    lcd.print(" UN VALOR MENOR ");

    delay (2000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Digit hora I _ R");
    lcd.setCursor(0,1);
    lcd.print("Digit hora F _ R");
    lcd.setCursor(13,0);
    lcd.print(numeroexacto1);
    lcd.setCursor(14,0);
    lcd.print(numeroexacto2);

    variable3= teclado.waitForKey();
  }
}

```

```

if( variable1)

{
// int inicioh;

// numeroexacto2 = variable2 -48;
numeroexacto3 = variable3 -48;
inicioh = numeroexacto1+numeroexacto2;

//  Serial.println("inicio");
//  Serial.println(inicioh);

Serial.print("Primer tecla ");
Serial.println(numeroexacto1);
Serial.print("Segunda tecla ");
Serial.println(numeroexacto2);
Serial.print("Tercer tecla ");
Serial.println(numeroexacto3);

int Valor = analogRead(0) / 4;
EEPROM.write(addressi,inicioh);

EEPROM.write(addressi+1,numeroexacto2);

EEPROM.write(addressi+2,numeroexacto3);

value= EEPROM.read(addressi);

value2= EEPROM.read(addressi+1);

valuef= EEPROM.read(addressi+2);

delay(3);
//  lcd.setCursor(13,0);
//  lcd.print(numeroexacto1);
//  lcd.setCursor(14,0);
//  lcd.print(numeroexacto2);
  lcd.setCursor(13,1);
  lcd.print(numeroexacto3);
  delay(3000);
  lcd.clear();

  lcd.setCursor(0,0);
  lcd.print("New hora programada");
  lcd.setCursor(0,1);
  lcd.print("En el riego");
  delay(2000);
  lcd.clear();

```

```

lcd.setCursor(0,0);

    int inicioh;
    inicioh = numeroexacto1+numeroexacto2;
    lcd.print(String("Inicia Cada ")+inicioh+"H"));
    lcd.setCursor(0,1);
    lcd.print(String("Termina Cada ")+numeroexacto3+"H"));
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,1);

}

}
else
{
    //{
    //Serial.println(value);

    // Serial.println(numeroexacto1);
    Serial.print("Son las:");
    DateTime now = RTC.now();
    /* Imprimimos el dia */
    lcd.setCursor(0,0);
    Serial.print(now.day(), DEC);
    Serial.print('/');
    lcd.print(now.day(), DEC);
    lcd.print('/');
    /* Imprimimos el mes */
    Serial.print(now.month(), DEC);
    Serial.print('/');
    lcd.print(now.month(), DEC);
    //lcd.print('/');
    /* Imprimimos el año */
    Serial.print(now.year(), DEC);
    Serial.print('/');
    //lcd.print(now.year(), DEC);
    /* Imprimimos la hora */
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    /* Imprimimos los minutos */
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    /* Imprimimos los segundos */
    Serial.print(now.second(), DEC);
    Serial.println();
    /* Comprobamos la hora y la enviamos por el puerto serie cada segundo */
    delay(1000);
    lcd.setCursor(5,0);

```

```

    /* icono */
    byte cwr[8]={
    B00001,
    B00100,
    B01001,
    B10001,
    B10001,
    B01000,
    B00100,
    B00001,
    };
    lcd.createChar(5,cwr);
    lcd.setCursor(8,0);
    lcd.write(5);
    byte relojd[8]={
    B10000,
    B00100,
    B00010,
    B00001,
    B11101,
    B00010,
    B00100,
    B10000,
    };
    lcd.createChar(6,relojd);
    lcd.setCursor(9,0);
    lcd.write(6);

    lcd.setCursor(9,1);

    lcd.print(String(now.hour()+":"+String(now.minute())+":"+String(now.second())));
    /* variables para representar las horas, minutos y segundo */
    hora =(now.hour());
    minuto=(now.minute());
    segundo=(now.second());

    value= EEPROM.read(addressi);
    //Serial.print("tecla - 1_ = ");
    Serial.print(value);
    Serial.println();

    value2 = EEPROM.read(addressi+1);
    Serial.print("Hora I - Cada = ");
    Serial.println(value2);
    //
    valuef = EEPROM.read(addressi+2);
    Serial.print("Hora F - Cada = ");
    Serial.println(valuef);

```

```

// int cas = ((value)(value2));
// Serial.print("cas"+cas);
// int Su = 3;
// int g = cas + Su;
// Serial.println(g);
//
int incremento = inicioh;
/* Incremento del valor Hora */
// Serial.print(incremento);
for (int i=0; i<=60; i=i+incremento)

{

if((minuto==i)&&(segundo == 1)||(minuto==i)&&(segundo == 2) )

{
    lcd.setCursor(6,0);
    lcd.print(value);

    /* imprime la hora si es hora de riego*/

Serial.println(String(now.hour()+":"+String(now.minute()+":"+String(now.secon
d()));
    lcd.setCursor(10,0); //posicion lcd Es Hora de Riego
    lcd.print(String(now.hour()+":"+String(now.minute()+":")+":");
    lcd.setCursor(0,1);
    Serial.println(" Es hora de Riego");
    lcd.print("Hora I R");
    digitalWrite (EVpin, HIGH);
    /* Encendido de Bomba */
dataString += String("Inicio RSD");
dataString += ",";
dataString
String(String(now.day()+"/"+String(now.month()+"/"+String(now.year()));
dataString = ",";
dataString
String(String("Inicio
RSD")+","+String(now.day()+"/"+String(now.month()+"/"+String(now.year()+","
+String(now.hour()+":"+String(now.minute()+":"+String(now.second()+","));

File dafile = SD.open("Encendido.txt",FILE_WRITE);

dafile.print(",");
dafile.println (dataString);
dafile.close();
Serial.println(dataString);
}
else

```

```

{
  valuef = EEPROM.read(addressi+2);

  int F = i+valuef;
  F;
  if((minuto ==F)&&(segundo ==0))//||(minuto ==F)&&(segundo ==1))
  {
    lcd.clear();
    lcd.setCursor(6,0);
    lcd.print(valuef);
    /* imprime la hora si finaliza hora de riego*/
    Serial.println(F);
  //    Serial.println(numeroexacto2 );

  Serial.println(String(now.hour()+":"+String(now.minute()+":"+String(now.secon
d()));
    lcd.setCursor(10,0);
    lcd.print(String(now.hour()+":"+String(now.minute())+"F");
    lcd.setCursor(0,1);
    Serial.println("No es hora de Riego");

    lcd.print("Hora F R");
    digitalWrite (EVPin, LOW);
    /* Apagado de Bomba */

  dataString += String("Fin RSD");
  dataString += ",";
  dataString +=
  String(String(now.hour()+":"+String(now.minute()+":"+String(now.second()));
  dataString += ",";

  /* Reporte del riego guarda en la memoria Sd*/

  File dafile = SD.open("reporte.txt",FILE_WRITE);

  dafile.print(",");
  dafile.println (dataString);
  dafile.close();
  Serial.println(dataString);

  }
  }
  }
  }
  }
}

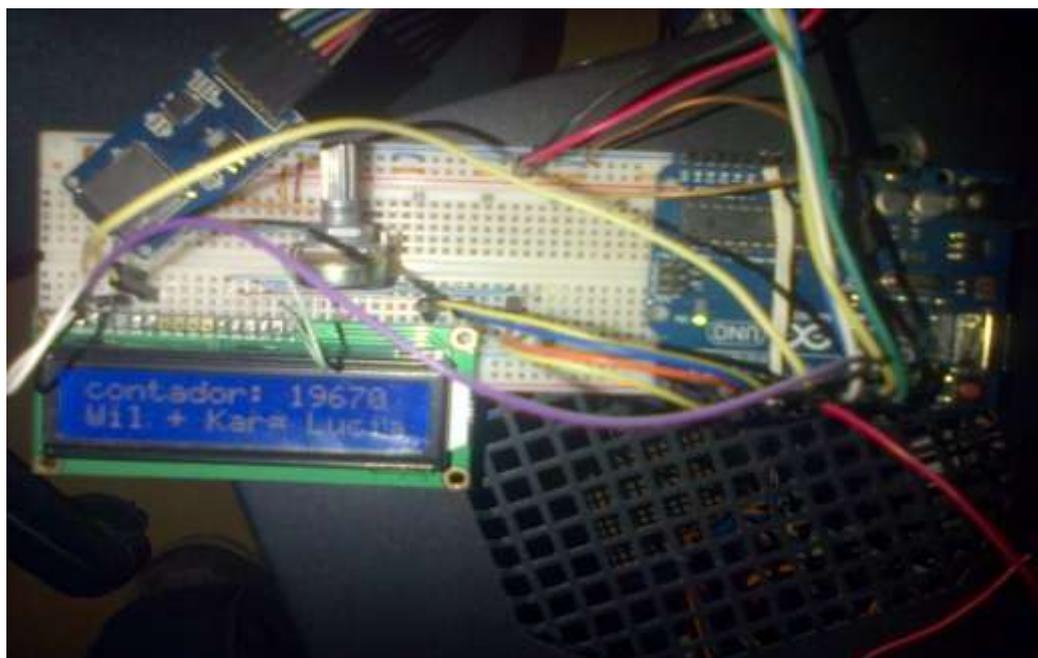
```

ANEXO 5: FOTOS DE LAS PROTOTIPOS DE PRUEBA DE LA PLACA ARDUINO Y SUS COMPONENTES

PROTOTIPO VERSIÓN 0.3.

Grafico # 1

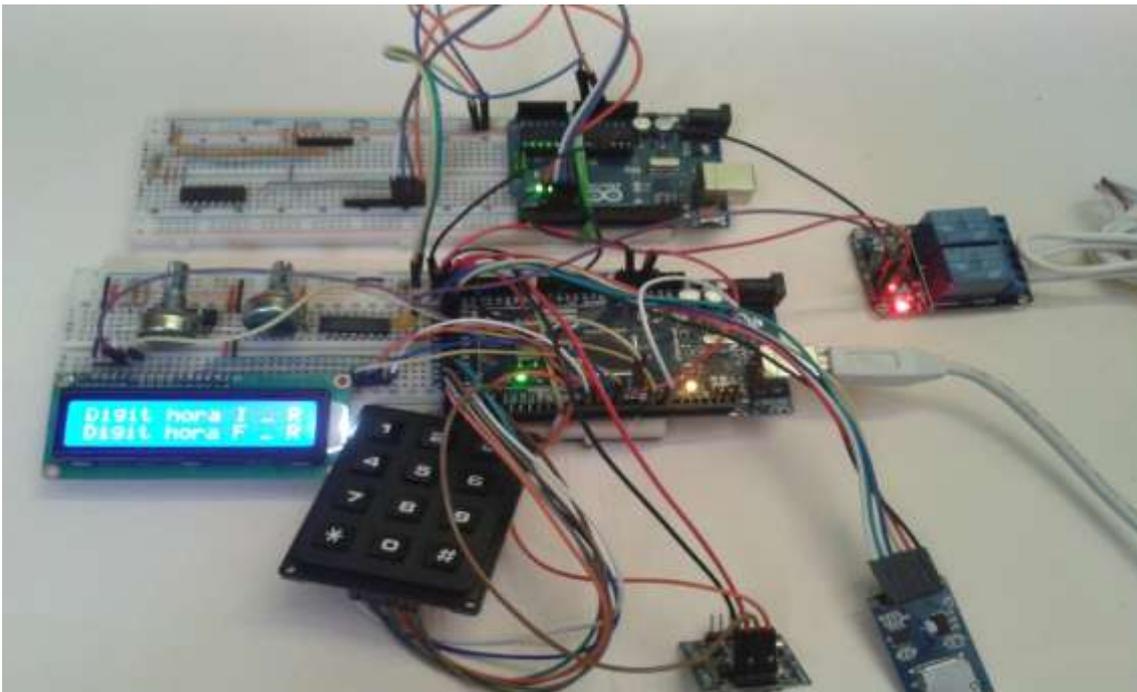
En esta etapa se revisó el uso adecuado del lector Sd y el Lcd 16x2 con la placa Arduino uno R3.



PROTOTIPO VERSIÓN 1.0.1.

Grafico # 2

En esta etapa se implementó el módulo de teclado, y el Arduino Mega ya que se necesitó de mayores puertos, para poder personalizar el horario del riego.



PROTOTIPO VERSIÓN 1.1

Grafico # 3

En esta etapa se implementó todos los módulos con placa Arduino a la placa con el circuito,

