

PROTOTIPO DE UN SISTEMA AUTOMATIZADO DE RIEGO PARA JARDINES

DIANA CAÑÓN QUIROGA
EDNA LUZ CIFUENTES CAMACHO

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA Y CIENCIAS BASICAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ, D. C.

2018

PROTOTIPO DE UN SISTEMA AUTOMATIZADO DE RIEGO PARA JARDINES

DIANA CAÑÓN QUIROGA
EDNA LUZ CIFUENTES CAMACHO

TRABAJO DE GRADO
PARA OPTENER EL TÍTULO DE INGENIERO DE SISTEMAS

DIRECTOR
CELIO GIL AROS
Ingeniero de Sistemas

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA Y CIENCIAS BASICAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ, D. C.

2018

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Bogotá D.C., 2 de Abril de 2018

DEDICATORIA

Este gran esfuerzo lo dedicamos a nuestros hijos que son el motor de nuestros sueños por quienes miramos al horizonte y creemos en nuestras capacidades aunque pasemos por momentos difíciles entendemos que este sacrificio es con el ánimo de saber que podemos dar más.

A nuestras familias quienes son la fuente de motivación e inspiración para seguir día a día con la fortaleza y la convicción de lograr una meta.

A nuestros compañeros de cada semestre que de alguna forma compartieron conocimientos, experiencias, alegrías y tristezas, con quienes tuvimos gratos momentos e intensos momentos de angustia pero con quienes finalmente logramos muchos objetivos dejando una huella positiva.

Sin olvidar que la Fundación Universitaria los Libertadores y nuestros docentes hacen parte fundamental para esta dedicatoria en donde la sabiduría y la reflexión son el fruto de lo que hoy mostramos con orgullo de lo que sembraron.

AGRADECIMIENTOS

Para todos aquellos que han hecho parte de este proceso y que de alguna forma nos aportan tiempo, esfuerzo, sabiduría y dedicación agradecemos su incondicional apoyo.

Por su sabiduría y dedicación a los docentes de ingeniería Ingeniero Celio Gil, Ingeniero Luis Eduardo Baquero, Ingeniero Miguel Hernández y demás docentes que hicieron parte de este proceso de formación.

Por su tiempo y esfuerzo a nuestros familiares quienes nos aportan valores, principios y sobretodo nos llenaron de energía en cada momento.

CONTENIDO

RESUMEN	1
INTRODUCCIÓN.....	2
1. ASPECTOS DE LA INVESTIGACIÓN	4
1.1. DESCRIPCIÓN DEL PROBLEMA.....	4
1.2. FORMULACIÓN DE LA PREGUNTA DE INVESTIGACIÓN.....	5
1.3. JUSTIFICACIÓN DEL PROYECTO DE INVESTIGACIÓN.....	5
1.4. IMPACTO.....	8
1.5. DELIMITACIÓN Y ALCANCE.....	8
1.6. RECURSOS.....	11
1.7. OBJETIVOS	13
1.7.1. OBJETIVO GENERAL.....	13
1.7.2. OBJETIVOS ESPECÍFICOS.....	13
1.8. LINEA DE INVESTIGACIÓN	14
2. MARCO TEORICO	15
2.1. ESTADO DEL ARTE.....	15
2.1.1. Antecedentes.....	15
2.1.2. Históricos.....	16
2.1.3. Productos similares.....	16
2.1.4. Tipos de riego	17
2.2. BASES TEORICAS.....	21
2.2.1 Lenguajes de programación.....	21
2.2.2 Bases Datos.....	21
2.2.3 PHPmyAdmin.....	22
2.2.4 Servicios de WebHost.....	22
2.2.6 Lenguaje unificado de modelado.....	22
2.2.7 Análisis y diseño orientado a objetos.....	23
2.2.8 Scrum metodología ágil	23
2.3. METAS A ALCANZAR	24
2.4. PRODUCTOS A ENTREGAR.....	24
2.5. DEFINICION DE TERMINOS BASICOS - GLOSARIO	25
3. DISEÑO METODOLOGICO	29

3.1.	METODOLOGIA DE INVESTIGACIÓN.....	29
3.1.1.	Tipo de Investigación	29
3.2.	METODOLOGIA DE DESARROLLO DE SISTEMA.....	29
3.2.1.	Historias de Usuarios	31
3.3.	DISEÑO DEL SISTEMA.....	34
3.2.	DISEÑO ARQUITECTONICO.....	39
3.3.	DISEÑO INTERFACE.....	40
3.4.	DISEÑO DEL CIRCUITO	41
3.5.	DISEÑO DE SEGURIDAD Y CONTROLES	41
3.6.	SELECCIÓN DE LA HERRAMIENTA DE DESARROLLO	43
3.1.6.	Codificación.....	43
4.	ANALISIS DE RESULTADOS Y CONCLUSIONES	46
4.1.	PRUEBAS	47
4.1.1.	Pruebas de Función.....	47
4.3.	INFORME DE PRUEBA	55
5.3.	CONCLUSIONES	57
5.4.	RECOMENDACIONES.....	58
	BIBLIOGRAFIA	59
	ANEXOS	60

TABLAS

Tabla 1 Cronograma de Actividades	10
Tabla 2 Recursos de Hardware.....	11
Tabla 3 Recursos de Software.....	12
Tabla 4 Recurso Humano	12
Tabla 6 Matriz de Funciones vs Usuarios.....	42
Tabla 7 Modulos.....	47
Tabla 8 Módulo Inser User. Prueba Caja Blanca.....	48
Tabla 9 Consultar datos. Prueba Caja Blanca.....	48
Tabla 10 BD_User Prueba caja blanca	49
Tabla 11 Prueba Caja Negra.....	50
Tabla 12 Informe Interfaz Usuario.	55
Tabla 13 Informe Interfaz de Ingreso.	55
Tabla 14 Informe Interfaz de Conexión.....	55
Tabla 15 Informe General.....	56

ILUSTRACIONES

Ilustración 1 Sistema de riego automatizado para jardines y huertos.....	17
Ilustración 2 Riego por goteo	18
Ilustración 3 Riego por Aspersión.....	20
Ilustración 6 Arduino Mega.....	25
Ilustración 7 Sistema de Riego.....	26
Ilustración 8 Sensor de humedad.....	26
Ilustración 9 Sensor de Temperatura.....	27
Ilustración 10 Reloj en tiempo real.	27
Ilustración 11 Relé.....	28
Ilustración 12 Modulo Bluetooth.....	28
Ilustración 16 Diagrama de Caso de uso.....	34
Ilustración 17 Diagrama de Clases sistema de riego.....	35
Ilustración 18 Diagrama de Secuencias sistema de riego.....	36
Ilustración 19 Diagrama de Actividades.....	37
Ilustración 20 Diagrama de componentes.....	38
Ilustración 21 Diagrama de Despliegue.....	38
Ilustración 22 Interfaz de Usuario Validación.....	40
Ilustración 23 Interfaz de Ingreso.....	52
Ilustración 24 Interfaz Menú principal.....	52
Ilustración 25 Interfaz Sensor de Humedad.....	53
Ilustración 26 Interfaz Sensor de Temperatura.....	54
Ilustración 27 Interfaz Consolidado de Riego.....	54

RESUMEN

Los sistemas de riego hacen más fácil una labor tan importante como la es la hidratación de las plantas estos sistemas controlan la cantidad y hora en la que se debe realizar dicha tarea. Este tipo de sistemas son usados tanto en cultivos macros como en pequeños jardines ubicados ya sea en patios, interiores o balcones.

Este proyecto está diseñado para pequeños jardines y su actividad puede ser monitoreada desde una aplicación móvil. Se construyó un pequeño jardín que simula cualquiera que se pueda tener en el hogar, se realizó uso de una protoboard, un sensor de humedad, un sensor de temperatura LM35, una tarjeta arduino y otros elementos capaces de interactuar para lograr un sistema integrado funcional. La aplicación que controla el prototipo está codificada en el lenguaje Arduino y capaz de avisar al sistema cuando debe activar su función de riego, la visualización de los datos capturados como lo son temperatura y humedad vs tiempo se reflejan en una aplicación que está diseñada en el lenguaje Android que envía y realiza consultas a una base de datos MySQL

Palabra Claves, Arduino, sensor de humedad, bluetooth, riego, Android.

INTRODUCCIÓN

En Egipto y Mesopotamia hace unos 4000 años, los egipcios y el pueblo de Mesopotamia, utilizaba una forma pasiva de riego, totalmente en función de la inundación anual del río. Los egipcios utilizaron el río Nilo como fuente de agua e hizo las cuencas individuales de los cultivos que entran en contacto con el agua de las inundaciones.

En Asia hace unos 2000 años, China usó los métodos tradicionales de riego por superficie y se fue a un nivel totalmente nuevo de riego, construcción de canales para la adquisición de agua desde muy lejos. La ciudad que ahora es Camboya, ha tenido un complejo sistema de canales, estanques, embalses para el riego y el almacenamiento de agua desde el siglo IX y XIV.

En Europa los romanos habían construido complejos sistemas de irrigación más de 2000 años atrás. Algunos canales transportaban agua de las montañas y lo depositó en los embalses. Local sistemas de ríos de agua de riego también se han descubierto. (START, 2017)

Una vez con la era moderna de vapor y las bombas de propulsión eléctrica, sacar agua de los ríos y utilizarla para regar las tierras. El sistemas de riego por goteo son hoy en día el más eficaz y eficiente de todos. Además, los sistemas de aspersión que utilizan el agua tanto como sea necesario son los sistemas más utilizados.

Los diferentes avances tecnológicos apoyan la iniciativa y necesidades de la sociedad, permitiendo la automatización de actividades cotidianas haciendo uso de las tecnologías de la información y de las comunicaciones, para tal efecto se plantea la implementación de un **Prototipo Sistema Automatizado de Riego** que permite generar un control de la humedad

para un jardín. Esto permite saber el momento adecuado en el que se deben regar las plantas evitando así que estén demasiado húmedas o demasiado secas.

El proyecto puede ser implementado para jardines pequeños e invernaderos ya que utilizan una misma lógica, inicialmente solo se realizará para jardines pequeños.

1. ASPECTOS DE LA INVESTIGACIÓN

1.1. DESCRIPCIÓN DEL PROBLEMA

Tener un jardín o algún tipo de cultivo requiere de ciertas atenciones como lo son: podar las plantas, o lo más necesario, regarlas. Todas estas tareas son habituales cuando las personas se dedican a la jardinería o a tener algún cultivo orgánico, pero cuando el usuario por falta de tiempo o por algún viaje que realice no puede cumplir con estas tareas las plantas mueren o simplemente tienden a enfermarse y dejar de dar frutos o florecer.

El inadecuado manejo de la temperatura o la humedad en el ambiente puede dar lugar a enfermedades abióticas que van desde deshidratación hasta congelamiento, además de esto pueden presentarse otras enfermedades como hongos que deterioran las plantas.

Este prototipo no es capaz de alertar al usuario sobre condiciones en las plantas diferentes a las del estado de humedad de su ambiente, tampoco sobre enfermedades de las mismas.

Es generar una solución a la problemática que se presenta con las personas que tienen, jardines, cultivos o grandes invernaderos controlando así la protección de los mismos y garantizando controles de temperatura y humedad capaces de mitigar el daño de las plantas o cultivos, aprovechando la tecnología como interfaz y mecanismo de seguimiento y monitoreo.

El agua a pesar de ser un recurso renovable y aparentemente abundante en el planeta, es un bien escaso que se agota, por tal motivo, su utilización para la práctica de los riegos en jardinería debe ser llevada a cabo de la forma más eficiente posible, evitando un consumo excesivo.

Se prevé que en 2030 el mundo tendrá que enfrentarse a un déficit mundial del 40% de agua en un escenario climático. (WRG, 2009)

La cuestión es que hay agua suficiente como para satisfacer las necesidades crecientes del mundo, pero no si no cambiamos radicalmente el modo en que se usa, se maneja y se comparte el agua. La crisis hídrica mundial es una crisis de gobernanza mucho más que de recursos disponibles. (WWAP, 2015)

1.2. FORMULACIÓN DE LA PREGUNTA DE INVESTIGACIÓN

¿Cómo a través de una aplicación móvil se puede monitorear y controlar el sistema automatizado de riego permitiendo minimizar el gasto de recursos naturales?

1.3. JUSTIFICACIÓN DEL PROYECTO DE INVESTIGACIÓN

Los avances tecnológicos se pueden mejorar, adaptar o construir para que un sistema sea más práctico, económico que genere un impacto beneficioso al ser humano y mostrase positivamente al medio ambiente.

Durante los siglos XVII y XVIII las familias reales europeas de París, Viena y Madrid, fueron las principales promotoras de la creación de jardines con un valor a la vez decorativo y científico, promoviendo la aclimatación de especies exóticas y la profundización en el estudio de la ciencia botánica. (biblioteca.ucm.es, 2012)

Este sistema da garantía de mantener las plantas con la cantidad suficiente de agua, la temperatura requerida de acuerdo con el jardín o cultivo.

El riego es tan importante para las plantas o cultivos como para las personas, durante toda su vida requieren de cuidados especiales al igual que el ser humano necesita del consumo del agua, las plantas también, aunque en este aspecto hay que prestar especial atención a esta necesidad dado que algunas no requieren de un riego frecuente.

Si a una planta no se la riega convenientemente, ésta comenzará a marchitarse, puede sufrir caída de hojas o comienzan a marchitarse. En cambio sí se excede en el riego, las raíces pueden llegar a pudrirse, por ello es muy importante conocer la planta y sus necesidades, para que siempre cuente con su mejor aspecto.

El suelo está formado por agregados y poros, cuya distribución constituye su estructura. La cantidad, tamaño y comunicación de los poros determinan la capacidad de retención de agua y sobre todo su velocidad de infiltración. Sin embargo, la estructura puede verse alterada por diversos agentes externos (impacto de gotas de lluvia o riego, tráfico peatonal o rodado, etc.), e internos (tipo de sales predominantes), por lo que el mantenimiento de una buena estructura mantendrá una infiltración adecuada. (plantasyjardines.es, 2014)

Es así como la posible composición del suelo y su humedad requiere una medición y control con la posibilidad de realizar un riego exacto, protegiendo y mejorando este sistema con un recurso automatizado.

1.3.1. Razones Sociales

Orientado a prestar un servicio social a los amantes de las plantas y los dueños de cultivos con la oportunidad de reducir gastos y consumo de recursos naturales, mostrando ambientes increíbles con este novedoso sistema de riego.

1.3.2. Razones Económicas

La implementación de sistemas automatizados trae consigo una inversión inicial que favorece con el tiempo el gasto de dinero y recursos naturales, el prototipo en su primera versión estará implementado en un diseño que mostrará su funcionalidad.

El uso de elementos tecnológicos y herramientas como Arduino que permite de forma íntegra generar un producto para satisfacer las necesidades de los seres humanos muestra la viabilidad de llevar a cabo este sistema.

1.3.3. Razones Técnicas

Cada día el mundo va girando más entorno a la tecnología y a todos sus avances, por esta razón es necesario que tareas que son repetitivas puedan automatizarse haciendo así más fácil el día a día del ser humano. A Pesar de que existen numerosos métodos de riego la gente sigue usando el método convencional ya sea por costos o por falta de conocimiento.

Este sistema de riego automatizado busca que más personas puedan acceder a métodos más modernos independientemente del tamaño o tipo de cultivo que tengan.

1.3.4. Razones Metodológicas

Suministrar a los usuarios que tengan un jardín, una herramienta que lleve el control del riego de sus plantas evitando así desperdicios de agua o deshidratación de las mismas.

Actualmente existen algunos proyectos similares como lo son Jarduino y Garden (TPI, 2013, p. 1) dirigidos a sistemas de riego utilizando tarjetas arduino, están orientados a balcones y cuentan con un sistema de riego por goteo lo que hace más lenta su tarea y puede producir huecos dando lugar a encharcamientos y heterogeneidad en la tierra.

El Prototipo de un Sistema Automatizado de Riego es fácil de instalar y brinda al usuario la tranquilidad de que sus plantas están en óptimas condiciones ambientales, su sofisticado sistema hace que reciban la cantidad adecuada de agua y no sufran por encharcamientos o resequedad, sus sensores permiten al programa saber el momento adecuado para activar el sistema, evitando así desperdicios innecesarios de agua o que el usuario esté pendiente del funcionamiento de este.

1.4. IMPACTO

Mostrar de manera creativa y tecnológica un proyecto para aquellas personas que se benefician de sus cultivos o simplemente para aquellos que viven en ambientes adornados con jardines y sistemas de solarios en espacios que se pueden adaptarse en distintos entornos. De otra parte minimizar el consumo de agua validando la temperatura y la humedad de la tierra evitando así encharcamientos y gasto innecesario del recurso hídrico, de esta forma optimizamos el consumo de este recurso. Considerando también la eficiencia en el manejo avanzado de la tecnología y optimizando el tiempo en estas tareas que se ejecutaban de manera manual.

1.5. DELIMITACIÓN Y ALCANCE

El proyecto está diseñado para jardines con un área no mayor a 20 m² ya que la motobomba que se usa no es capaz de impulsar el agua con tal fuerza que abarque un área mayor, lo que produciría que el riego no se ejecute de manera homogénea.

El circuito electrónico enviará datos a la aplicación móvil correspondientes a las variables de humedad, temperatura y tiempo, además realizara esta acción cada vez que el circuito ejecute su función de riego, para que el usuario pueda tener una vista actualizada de la información enviada por el circuito, esta aplicación debe contar con conexión a internet y mantener activo el enlace a bluetooth desde el equipo seleccionado para verificar y controlar la información.

La aplicación móvil cuenta con dos tipos de usuario, un usuario general y un administrador que tiene acceso al módulo de editar usuarios. Los usuarios pueden visualizar la información registrada por cada sensor a través del microprocesador de la tarjeta Arduino, mostrando así un informe detallado de las gráficas generadas por la aplicación.

1.5.1. Espacial

La delimitación espacial se efectuará en las instalaciones de la Fundación Universitaria los Libertadores en la ciudad de Bogotá con el acompañamiento y la asistencia de los ingenieros del programa de Ingeniería de Sistemas y de Electrónica.

1.5.2. Cronológica

Este proyecto se ejecutará en un tiempo estimado de 7 meses calendario, iniciando desde el mes de septiembre 2017 hasta el mes de marzo de 2018.

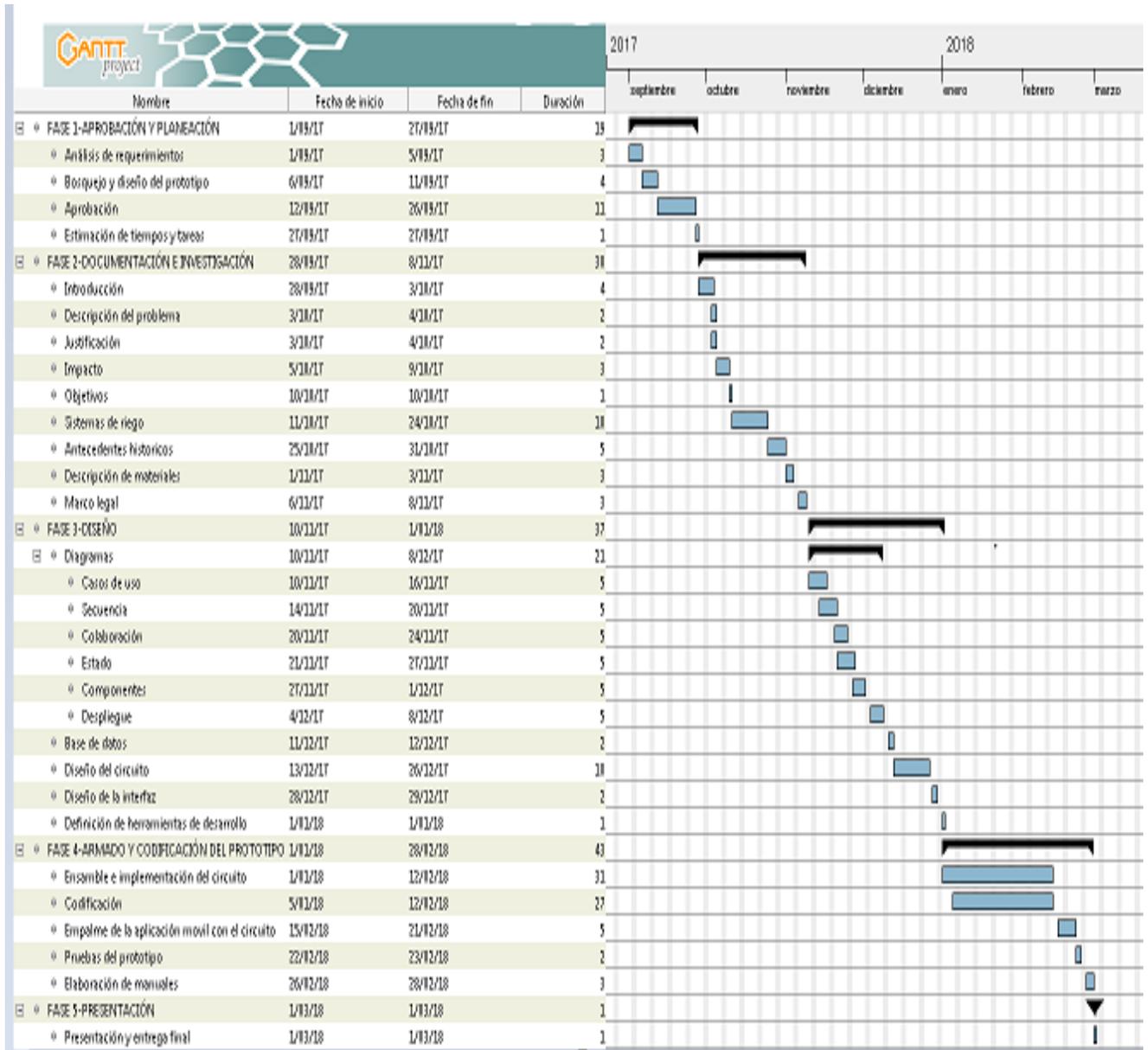


Tabla 1 Cronograma de Actividades

1.6. RECURSOS

Para la realización del presente proyecto se cuenta con los siguientes recursos:

Recursos de hardware

Descripción Recurso	Cantidad	Costo
Tarjeta Arduino Mega	1	56.000
Reloj en tiempo real.	1	12.000
Protoboard	1	10.000
Módulo Relé 12 V	1	7.000
Cables para circuito	1 m	400
Sensor de humedad para tierra	1	13.000
Módulo Bluetooth HC-05	1	19.000
Contenedores plásticos	2	20.000
Motobomba	1	30.000
Manguera Nivel 5/16" Calibre 20	2 m	2.000
Tubo PVC	3 m	8.500
Total		177.900

Tabla 2 Recursos de Hardware

Recursos de software

Descripción	Cantidad	Costo
Android MIT App Inventor	1	Licencia libre
Motor de Bases de datos: MySQL	1	Licencia libre
PHP	1	Licencia libre
ArgoUML	1	Licencia libre
Gantt Project	1	Licencia libre
Arduino	1	Licencia Libre
Google Charts		Licencia Libre

Tabla 3 Recursos de Software.

Recurso Humano

Recurso humano	Cantidad horas	Costo unitario	Costo total
Análisis del prototipo (Autores)	86	\$13.000	\$1.118.000
Diseño y ensamble de estructura de riego (Autores)	10	\$15.000	\$150.000
Diseño y armado del circuito (Autores)	24	\$20.000	\$480.000
Programación (Edna)	20	\$20.000	\$400.000
Pruebas (Autores)	6	\$10.000	\$60.000
Implementación (Diana)	10	\$15.000	\$150.000
Ensamble de componentes	3	\$25.000	\$75.000
Total			\$2.433.000

Tabla 4 Recurso Humano

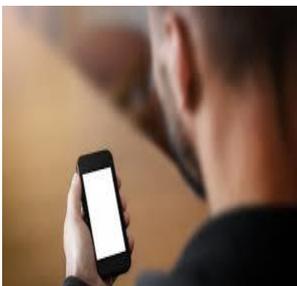
1.7.OBJETIVOS

1.7.1. OBJETIVO GENERAL

Diseñar e implementar un Prototipo de Sistema Automatizado de Riego para Jardines.

1.7.2. OBJETIVOS ESPECÍFICOS

- Diseñar una estructura base para la implementación del prototipo.
- Implementar el prototipo haciendo uso de arduino, sensores y circuitos electrónicos.
- Desarrollar la aplicación de software que se integra con arduino para la automatización del sistema de riego.
- Elaborar una réplica de un jardín para las pruebas de riego pertinentes.
- Simular el sistema de Riego de manera automática un jardín haciendo uso del prototipo.
- Visualizar gráficas estadísticas de temperatura y riego.



1.8. LINEA DE INVESTIGACIÓN

La línea de Investigación del programa Ciencias de la Información y las Comunicaciones tiene como objetivo principal estudiar a profundidad la relación entre la captura, generación, procesamiento, transferencia, almacenamiento, utilización y uso intensivo y extensivo de información producto de múltiples casos de uso usando tecnología.

Enfatizando en las siguientes iniciativas:

- Internet de las Cosas
- Aprendizaje de Maquina
- Ciencia de Datos
- Estadística aplicada

Esta línea se enmarca en la Línea Institucional “Sistemas Complejos y Aplicaciones tecnológicas de Impacto Social”

El proyecto desarrollado desde su objetivo principal contribuye al fortalecimiento de la línea de investigación del programa a través del desarrollo de aplicaciones móviles realizando uso de herramientas que favorecen y facilitan el uso de las mismas.

2. MARCO TEORICO

2.1. ESTADO DEL ARTE

2.1.1. Antecedentes

Sistemas de riego: Los primeros registros del riego en agricultura se remontan al año 6000 a.C. en Egipto y en Mesopotamia (Irak e Irán en la actualidad) cuyos pobladores utilizaban los patrones de riada del Nilo o del Tigris y Éufrates, respectivamente. Las inundaciones que ocurrían de julio a diciembre, eran desviadas hacia los campos durante unos 40 ó 60 días. Luego se drenaba el agua hacia el río en el momento preciso del ciclo de cultivo.

Un segundo diseño consistiría en una serie de escaleras descendiendo en el río.

Cuatro siglos después, en la primera dinastía de Egipto, se construyó el primer proyecto de riego a gran escala, bajo el reinado del rey Menes. Se utilizaron presas y canales para dirigir las aguas de inundación del Nilo hacia el lago Moeris.

Sería ya un milenio más tarde cuando aparecieron las tuberías de cemento y de roca molida. Los famosos acueductos, una invención construida por los ingenieros romanos, permitían transportar el agua salvando los desniveles del terreno.

Por otro lado, en la cultura azteca destacó el conocido como cultivo por chinampas, que consistía en una construcción de campos elevados dentro de una red de canales dragados sobre el lecho del lago. Así se reciclaban los nutrientes arrastrados por las lluvias. Los mayas, que estaban asentados en la selva tropical, establecieron diferentes técnicas adecuadas para cada tipo de terreno: campos elevados en zonas inundables y terrenos con desnivel en zonas de excesiva humedad. Construían terrazas de cultivo sostenidas por muros, así podían modificar la pendiente del terreno, contribuyendo a preservar la humedad y a mejorar la fertilidad del suelo. (Traxco, 2017).

2.1.2. Históricos

El rey babilónico Hamurabi, autor del famoso código jurídico, fue quien elaboró las primeras regulaciones sobre el *agua*. Tenía en consideración la distribución del agua de una manera proporcional, con base en la superficie labrada, también se definía la responsabilidad del agricultor de realizar mantenimiento de los canales de propiedad, así como la administración colectiva del canal por parte de todos sus usuarios. (Traxco, 2010)

En este punto de la historia, el desarrollo agrícola radicaba en una serie de técnicas para manejar el agua del riego a través de los sistemas de distribución y en la construcción de terrazas de cultivo. Se desarrollaban tecnologías enfocadas a mitigar los efectos de la erosión, aminorar las inundaciones, retener humedad, y permitir captaciones, traslados y almacenamientos.

2.1.3. Productos similares

- **Invernadero de Riego y temperatura controlado remotamente desde la Web basado en tecnologías móviles.**

Proyecto Invernadero ejecutado en la Universidad los Libertadores controlado remotamente desde la Web basado en tecnologías móviles realizando uso de Raspberry PI2 y desarrollado en Python 2.7. Con el objetivo de controlar la temperatura dentro de un invernadero y lograr monitorear desde la Web. (Barón Alarcón, 2017)

- **Sistema de riego automatizado para jardines y huertos de pequeño y gran tamaño:**

Un sistema de riego automatizado como el de Karcher cuenta con un sensor de lluvia que registra las precipitaciones que se producen y las tiene en cuenta al momento de activarse de nuevo. Es decir, que el sistema sólo riega las plantas cuando éstas realmente lo necesitan. Con el sistema de riego automatizado de Karcher también se puede crear un

sistema de riego simulador de lluvia que permitirá distribuir el agua con aspersores en un área específica que se desee regar ya sea en su periferia o al interior. (ferrepat, 2016)



*Ilustración 1 Sistema de riego automatizado para jardines y huertos
Fuente: (<http://www.revista.ferrepat.com>, s.f.)*

2.1.4. Tipos de riego

- **Riego por goteo**

El riego por goteo o riego gota a gota es un método de irrigación que permite una óptima aplicación de agua y abonos en los sistemas agrícolas de las zonas áridas. El agua aplicada se infiltra en el suelo irrigando directamente la zona de influencia radicular a través de un sistema de tuberías y emisores.

Actualmente se han añadido varias mejoras en los emisores:

- **Goteros autocompensantes**

Estos emisores ofrecen un caudal fijo dentro de un rango más o menos amplio de presión. La utilidad de estos goteros radica en la capacidad de homogeneización del riego a lo largo de una línea de riego, ya que los últimos emisores de la línea normalmente tienen una menor presión que los primeros debido a la caída de presión por rozamiento del agua con la tubería.

- **Goteros antidrenantes**

Estos goteros se cierran automáticamente al bajar la presión en el sistema de riego, de manera que no ocurre la descarga de la tubería, lo que produce ventajas tales como evitar la entrada de aire al sistema y la bomba de riego no necesita cargar el sistema para empezar a funcionar, por tanto optimiza su uso. (Novedades-agrícolas, 2016)

- **Goteros regulables**

Permiten regular el caudal con un mando mecánico.



Ilustración 2 Riego por goteo
(<http://www.revista.ferrepat.com>, s.f.)

- **Riego por Aspersión**

Estos sistemas son los primeros que se desarrollaron en el riego por aspersión. Se dividen a su vez en:

Sistemas fijos

Se colocan los aspersores en el marco establecido, y el sistema de tuberías puede ser enterrado o bien superficial, quedando como parte saliente y con la altura adecuada el vástago donde irá incorporado el aspersor.

Sistemas Semifijos

Son esencialmente sistemas que se van desplazando de una zona a otra de forma manual o mecanizada mediante un desmontaje rápido del sistema. Dentro de estos sistemas se encuentran las alas de riego y los cañones de riego.

Sistemas auto mecanizados

Son sistemas automotrices que llevan instalados motores eléctricos o sistemas hidráulicos que permiten su movimiento a lo largo de la superficie de riego. Dentro de estos sistemas se encuentran los sistemas pivotantes de riego, los sistemas de desplazamiento lateral (carros de riego) y otras máquinas regadoras.

La elección de un sistema u otro se establece según los criterios técnico-económicos que permiten o no su instalación, como son la superficie a regar, la orografía del terreno, el acceso a la electricidad en la finca, etc.

El uso de sistemas de irrigación por goteo está muy extendido en cultivos extensivos (cereales, forrajes, patata, remolacha, etc.) y hortícolas. Se utiliza en terrenos con orografía irregular, donde los recursos hídricos son abundantes.

Ventajas de la aspersión

- Tiene un menor consumo de agua que los sistemas de riego por inundación.
- Presenta una gran adaptabilidad a terrenos irregulares, con grandes diferencias de cota en su superficie.
- Permite dosificar el agua con una buena precisión.
- Su distribución sobre el material vegetal depende del viento, aunque a bajas velocidades es muy homogénea.

- Es utilizado para la aplicación de riego antihelada y la aplicación de fitohormonas. (novedades-agrícolas, 2016)



*Ilustración 3 Riego por Aspersión
([Http://www.mapama.gob.es](http://www.mapama.gob.es), s.f.)*

- **Riego por Microaspersión**

Los Microaspersores están destinados a suministrar el riego mediante gotas muy finas. Poseen un deflector giratorio, denominado rotor o bailarina, que ayuda a ofrecer un mayor diámetro de cobertura, una menor tasa de precipitación que los difusores, un mayor tamaño de gota, y una mejor distribución del agua (sobre todo en uniformidad de distribución). Por Cada tipo de microaspersor existen varios tipos de rotores (bailarinas).

La diferencia principal con la nebulización es que la microaspersión proyecta en agua en forma de chorros diminutos a la planta, en lugar de suministrarla en forma nebulizada, y a su vez disponen de elementos giratorios que distribuyen el agua en la superficie.

2.2. BASES TEORICAS

2.2.1 Lenguajes de programación

Los lenguajes de programación de una computadora en particular se conocen como código de máquinas o lenguaje de máquinas.

A diario se interactúa con distintos tipos de máquinas como celulares, tabletas y computadores. Todos ellos tienen un lenguaje de programación que ayuda a traducir las órdenes del usuario para que pueda utilizarlos.

Un lenguaje de programación es básicamente un sistema estructurado de comunicación, similar al humano, el cual nos permite comunicarnos por medio de signos, ya sean palabras, sonidos o gestos. Refiriéndonos a los aparatos, este sistema está organizado para que se entiendan entre sí y a su vez interprete las instrucciones que debe ejecutar. (Colombiadigital, 2016)

2.2.2 Bases Datos

Conjunto de informaciones que está organizado y estructurado de un modo específico para que su contenido pueda ser tratado y analizado de manera rápida y sencilla.

Las bases de datos, por lo tanto, presentan datos estructurados de acuerdo a diferentes parámetros. Al disponer la información de una cierta forma, el usuario puede encontrar aquello que busca con facilidad, a diferencia de lo que le sucedería si todos los datos estuvieran mezclados y sin ningún tipo de orden. (<https://definicion.de/base-de-datos/>)

MySQL es la base de datos de código abierto más popular del mundo. Con su rendimiento, confiabilidad y facilidad de uso comprobados, MySQL se ha convertido en la principal opción de base de datos para aplicaciones basadas en la Web, utilizada por propiedades web de alto perfil como Facebook, Twitter, YouTube, y los cinco principales sitios web*. Además, es una alternativa extremadamente popular como base de datos integrada, distribuida por miles de ISV y OEM. (ORACLE)

2.2.3 PHPmyAdmin

Es una herramienta que se ofrece desde los paneles de control cPanel de los alojamientos web de HOSTINET con la que podremos manejar y administrar nuestras bases de datos MySQL. Se pueden crear, eliminar, modificar bases de datos así como gestionar las tablas de las mismas. (Hostinet, 2016)

2.2.4 Servicios de WebHost

Es un alojamiento en la Web en el prototipo usamos el 000wehost File que nos provee un espacio en el servidor para alojar la información gratuita y lograr interactuar con la aplicación de forma independiente e interactiva sin preocupaciones y de forma económica. 000Webhost es uno de los hosting web más grandes de la red, especialmente en su modalidad gratuita. Esta plataforma ofrece a los usuarios 1.5GB de almacenamiento junto con 100 GB de tráfico al mes para alojar sus páginas web de forma totalmente gratuita. También ofrece una modalidad de pago con numerosas características adicionales. (<https://www.redeszone.net>)

2.2.6 Lenguaje unificado de modelado

(UML, por sus siglas en inglés, Unified Modeling Language). Las grandes aplicaciones empresariales, los que ejecutan aplicaciones de negocios, deben ser algo más que un grupo de módulos de código. Ellos deben estructurarse de una manera que permite la escalabilidad, seguridad, y la ejecución robusto bajo condiciones de estrés, y su estructura refieren con frecuencia como su arquitectura debe ser definido con suficiente claridad que los programadores de mantenimiento puedan rápidamente encontrar y corregir un error que aparece mucho después de que los autores originales se han trasladado a otro proyecto. Por supuesto, una arquitectura bien diseñada beneficia a cualquier programa, y no sólo a los más grandes, como hemos señalado aquí. Mencionamos grandes aplicaciones en primer lugar porque la estructura es una forma de lidiar con la complejidad, por lo que los

beneficios de la estructura (y de modelado y diseño, como demostraremos) compuesto como el tamaño de la aplicación aumenta de tamaño.

2.2.7 Análisis y diseño orientado a objetos

La metodología de Análisis y Diseño Orientado a Objetos se ha usado ampliamente en el desarrollo de aplicativos orientados a la simulación, y al mismo tiempo se ha convertido en la metodología estándar en la industria del software, considerada también como una de las mejores prácticas para desarrollar proyectos de software con calidad. (Pressman, 2001)

Debido a su sencillez, esta metodología abarca de manera muy general la estructura de las interfaces de software haciendo énfasis solamente en las entradas y salidas de cada módulo, sin entrar en detalles de cómo se almacenan las variables o estructuras de datos en cada procedimiento.

2.2.8 Scrum metodología ágil

Scrum es un proceso de la Metodología Ágil que se usa para minimizar los riesgos durante la realización de un proyecto, pero de manera colaborativa.

Entre las ventajas se encuentran la productividad, calidad y que se realiza un seguimiento diario de los avances del proyecto, logrando que los integrantes estén unidos, comunicados y que el cliente vaya viendo los avances.

En primer lugar se define el Product Backlog, lo que nos permitirá realizar los Sprint del proyecto. (Scrum en la Metodología Ágil, 2017)

2.3. METAS A ALCANZAR

Implementar el prototipo en un sistema funcional que permita dar cuenta de todo el proceso ingenieril aplicado a este tipo de proyectos en beneficio de la sociedad o usuarios que posean jardines ya sean de tamaño micro o macro.

- Metas a corto plazo: Se realiza el levantamiento de la información, delimitación del alcance, definición de requerimientos funcionales y no funcionales, por último la definición de objetivos alcanzables.
- Metas a mediano plazo: Se trabajara en el análisis y diseño del proyecto. También se darán los cimientos del prototipo del software. Desarrollo del Prototipo
- Metas a largo plazo: Realizar un entregable del producto teniendo en cuenta dos puntos importantes la calidad y los objetivos propuestos.

2.4. PRODUCTOS A ENTREGAR

- Estructura física para el riego
- Circuito con conexión para los sensores de humedad, temperatura, reloj en tiempo real, relé.
- Aplicativo web para estadísticas de riego.
- Base de datos.
- Requerimientos funcionales y no funcionales
- Diagramas de modelado unificado (UML)
- Código fuente.
- Manual técnico.

2.5.DEFINICION DE TERMINOS BASICOS - GLOSARIO

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 hobby y para cualquiera interesado en crear objetos o entornos interactivos.

El microcontrolador de la placa se programa usando el Arduino ProgrammingLanguage (basado en Wiring) y el Arduino DevelopmentEnvironment (basado en Processing).

Las placas se pueden ensamblar a mano o encargarlas pre ensambladas; el software se puede descargar gratuitamente.



*Ilustración 4 Arduino Mega
Fuente: Autores*

Sistema de riego

Se denomina Sistema de riego o perímetro de riego, al conjunto de estructuras, que hace posible que una determinada área pueda ser cultivada con la aplicación del agua necesaria a las plantas. El sistema de riego consta de una serie de componentes, aunque no necesariamente el sistema de riego debe constar de todas ellas, ya que el conjunto de componentes dependerá de si se trata de riego superficial (principalmente en su variante de riego por inundación), por aspersión, o por goteo. (wikipedia)



Ilustración 5 Sistema de Riego
Fuente: desarrollandoamerica.org

Sensor de humedad de suelo:

Este sensor de humedad puede leer el porcentaje de humedad presente en la tierra. Es un sensor de baja tecnología, pero es ideal para el seguimiento de un jardín urbano. Se trata de una herramienta indispensable para un jardín de contacto. Este sensor utiliza las dos sondas para pasar corriente a través de la tierra, y luego lee la resistencia para obtener el nivel de humedad. Más agua hace que el suelo conduzca la electricidad con mayor facilidad (menos resistencia), mientras que el suelo seco es un mal conductor de la electricidad (más resistencia).

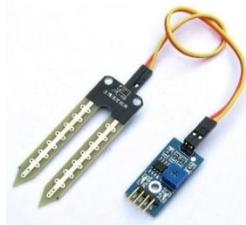


Ilustración 6 Sensor de humedad.
Fuente: Autores

Sensor de temperatura LM35:

El LM35 es un sensor de temperatura digital. A diferencia de otros dispositivos como los termistores en los que la medición de temperatura se obtiene de la medición de su resistencia eléctrica, el LM35 es un integrado con su propio circuito de control, que proporciona una salida de voltaje proporcional a la temperatura.



*Ilustración 7 Sensor de Temperatura
Fuente www.luisllamas.es*

Reloj en Tiempo real

Es un reloj de un ordenador, incluido en un circuito integrado, que mantiene la hora actual. Aunque el término normalmente se refiere a dispositivos en ordenadores personales, servidores y sistemas embebidos, los RTC están presentes en la mayoría de los aparatos electrónicos que necesitan guardar el tiempo exacto. (https://es.wikipedia.org/wiki/Reloj_en_tiempo_real)



*Ilustración 8 Reloj en tiempo real.
Fuente <http://k-electronica.es>*

Relé o relevador:

Es un dispositivo electromagnético. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes. Fue inventado por Joseph Henry en 1835.



Ilustración 9 Relé.

Fuente: Autores

Modulo Bluetooth HC 05:

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz. Los principales objetivos que se pretenden conseguir con esta norma son:

Facilitar las comunicaciones entre equipos móviles.

Eliminar los cables y conectores entre estos.

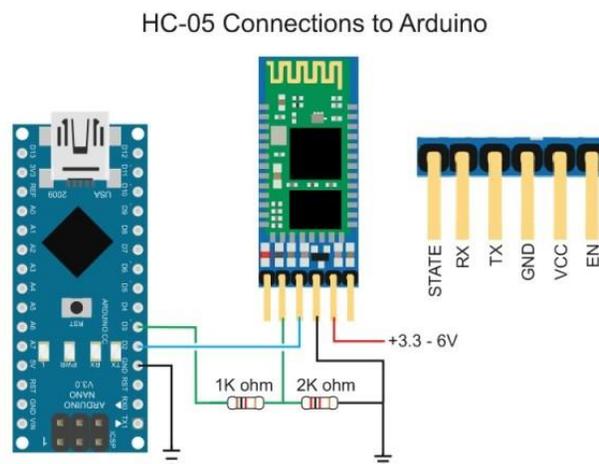


Ilustración 10 Modulo Bluetooth.

Fuente <http://www.geekbotelectronics.com>

3. DISEÑO METODOLOGICO

3.1. METODOLOGIA DE INVESTIGACIÓN

3.1.1. Tipo de Investigación

DESCRIPTIVA

Observamos las necesidades que se tienen en los hogares y en algunos solares buscando implementar una forma fácil de manipular un sistema de riego que muestra de forma cualitativa y cuantitativa las veces que ejecuta un riego, minimizando los recursos naturales, teniendo en cuenta factores del tiempo, humedad y de temperatura que lo hace eficiente siendo así un causante de la transformación tecnológica.

3.2. METODOLOGIA DESARROLLO DEL SISTEMA

Para el desarrollo del proyecto se seleccionó la metodología ágil SCRUM por ser un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto . Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales. (proyectosagiles.org, 2017)

Definición de Roles

Se ejecuta metodología con la intervención de dos Ingenieras quienes realizaran cada una los roles asignados de acuerdo a las fortalezas con el fin de desarrollar a conformidad las actividades planteadas:

Scrum Master Diana Cañon encargada de liderar el equipo con el objetivo de dar cumplimiento a los tiempos y actividades establecidas, verifica los requerimientos y hace la retrospectiva para que las cosas funcionen perfectamente y optimizar lo que se pueda mejorar.

El Product Owner Edna Cifuentes es la encargada de las funciones que están priorizadas según lo que es más y menos importante para el negocio como canal de comunicación.

El objetivo es que el Product Owner responda la pregunta “¿Qué hay que hacer?”.¹ Definimos los objetivos y los alcances del producto que se va a desarrollar priorizando las necesidades.

Team Scrum el equipo conformado por Edna Cifuentes y Diana Cañon con los objetivos claros para la ejecución del proyecto desarrollando cada iteración con calidad

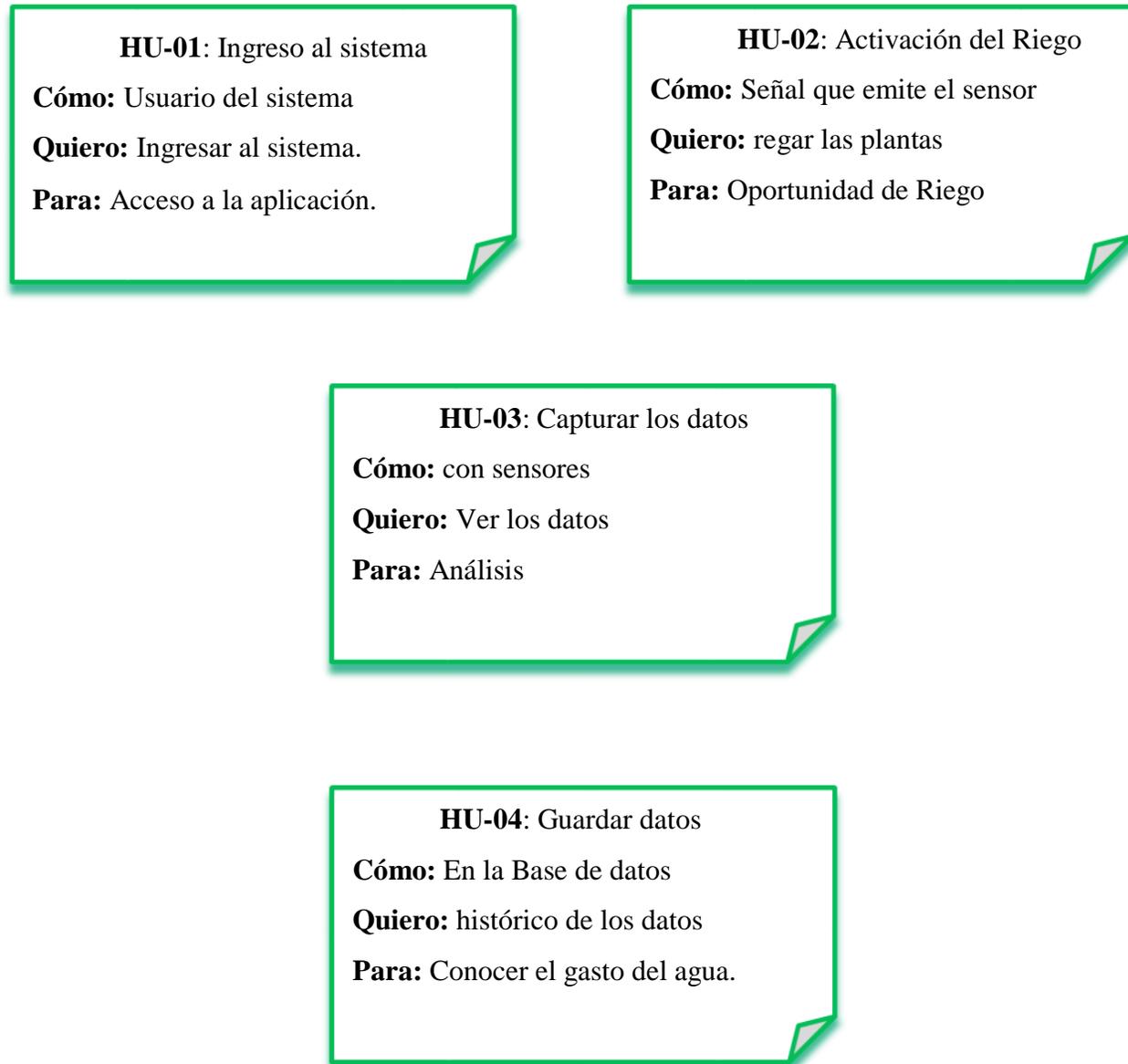
Definidos los roles de cada una de las participantes al desarrollo del proyecto realizamos el cronograma de reuniones semanales para definir los Product backlog.

Product backlog: Es una “wish list” sobre las funcionalidades del producto. Es el conjunto de requisitos denominados en el proceso como historias de usuario, descritos en un lenguaje sin tecnicismo en donde se prioriza el ROI (retorno de la inversión) considerando su costo beneficio, durante la ejecución del proyecto se van realizando los ajustes a los requisitos y prioridades en intervalos de tiempo regulares. (Engineers, 2013)

¹ (SCRUM, 2014).

El Product Owner al reunirse con su equipo de trabajo realiza la redefinición de las Historias de Usuario como se evidencian en las siguientes gráficas.

3.2.1. Historias de Usuarios



HU-05: Crear Usuarios.**Cómo:** Usando la aplicación.**Quiero:** Permitir acceso a otros.**Para:** Acceso a la aplicación.**HU-06: Modificar Usuarios.****Cómo:** Usando la aplicación.**Quiero:** Actualizar Datos.**Para:** Tener usuarios activos.**HU-07: Resultados.****Cómo:** con sensores.**Quiero:** Ver gráficas.**Para:** Controlar el sistema en tiempo

Sprint backlog: Es un subconjunto de ítems del Product Backlog, que son seleccionados por el equipo para realizar durante el Sprint sobre el que se va a trabajar. El equipo establece la duración de cada Sprint.². Los avances de cada una de las fases que presentados al docente, en donde se sugieren los cambios pertinentes.

² (<https://www.softeng.es>)

Los siguientes Sprint estarán especificados en los anexos de la siguiente forma:

Sprint Backlog			
Sprint	1	Planeación	Anexo A
Sprint	2	Documentación e Investigación	Anexo B
Sprint	3	Diseño Estructural	Anexo C
Sprint	4	Codificación	Anexo D
Sprint	5	Ajustes	Anexo E
Sprint	6	Pruebas	Anexo F
Sprint	7	Verificación	Anexo G
Sprint	8	Documentación	Anexo H

Sprint planning meeting: Esta reunión se hace al comienzo de cada Sprint y se define cómo se va a enfocar el proyecto que viene del Product Backlog las etapas y los plazos. Cada Sprint está compuesto por diferentes features.

Daily scrum o Stand-up meeting: Es una reunión breve que se realiza a diario mientras dura el periodo de Sprint. Se responden individualmente tres preguntas: ¿Qué hice ayer?, ¿Qué voy a hacer hoy?, ¿Qué ayuda necesito?³ Realizamos reuniones parciales y nos conectamos remotamente para aclarar dudas y apoyar las necesidades.

Sprint review: Se revisa el sprint terminado, y ya debería haber un avance claro y tangible para presentárselo al cliente.⁴ Reuniones periódicas que nos garantiza la verificación de los avances.

³ (<https://openwebinars.net>, 2014)

⁴ (Softeng, 2015)

3.3. DISEÑO DEL SISTEMA

✓ Diagrama de Caso de Uso

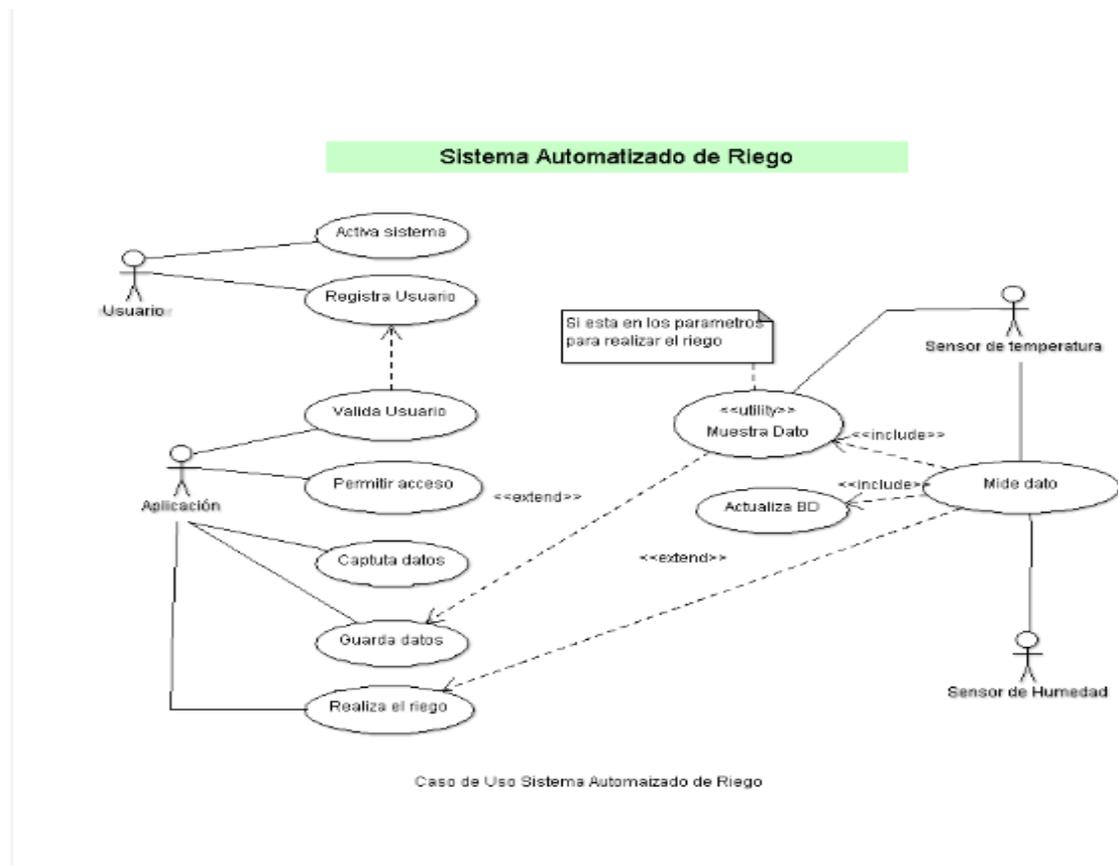
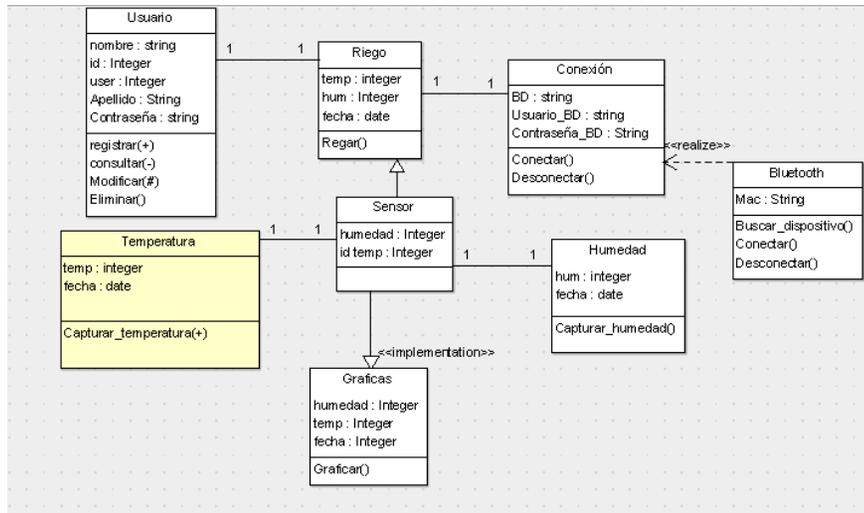


Ilustración 11 Diagrama de Caso de uso.

Fuente: Autores

✓ Diagrama de Clases



*Ilustración 12 Diagrama de Clases sistema de riego.
Fuente: Autores.*

✓ Diagrama de Secuencia

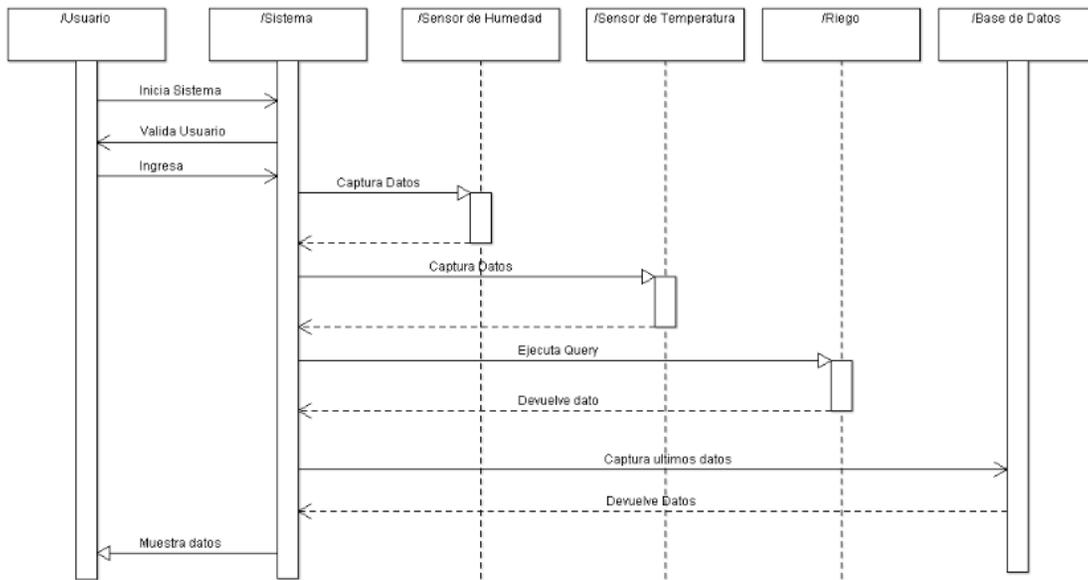
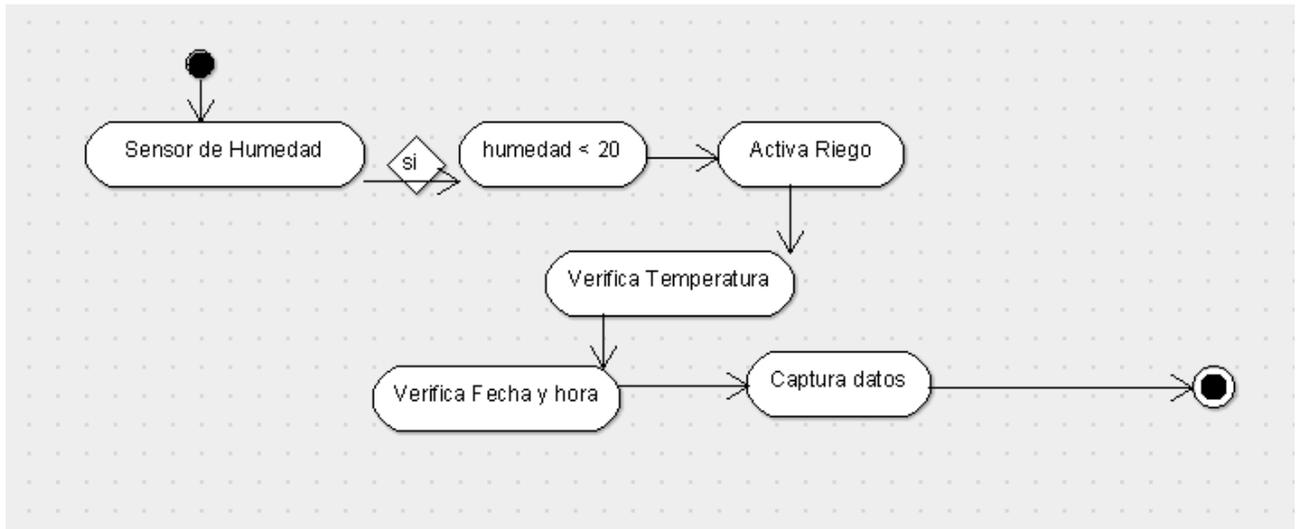


Diagrama de Secuencias Sistema de Riego Automatizado

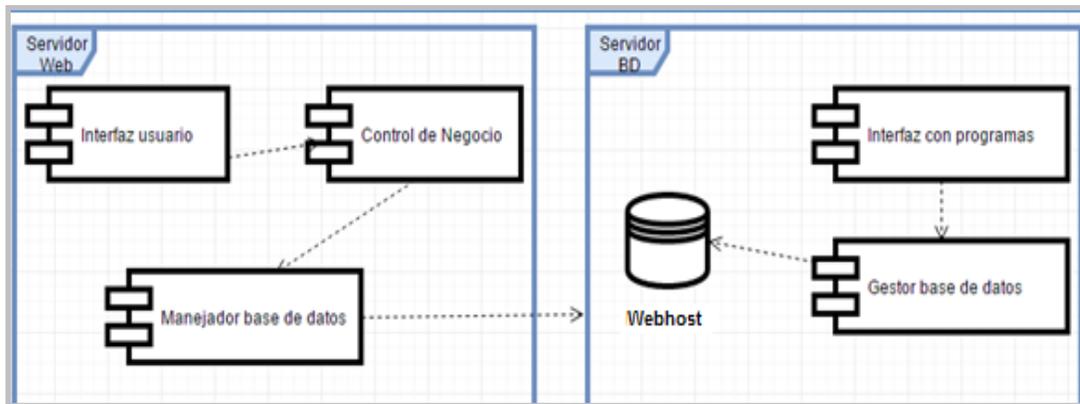
*Ilustración 13 Diagrama de Secuencias sistema de riego.
Fuente: Autores.*

✓ Diagrama de Actividades



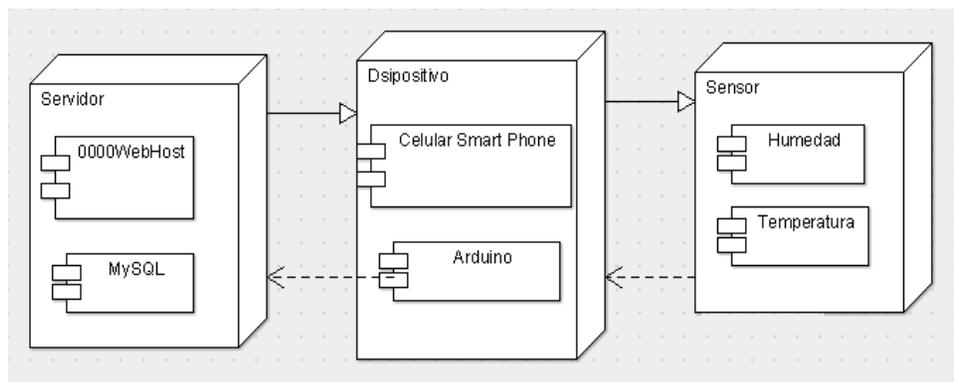
*Ilustración 14 Diagrama de Actividades.
Fuente: Autores.*

✓ Diagrama de Componentes



*Ilustración 15 Diagrama de componentes.
Fuente: Autores.*

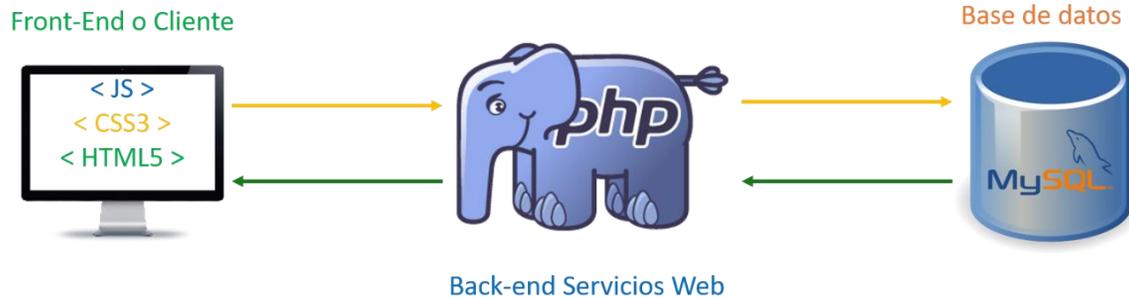
✓ Diagrama de Despliegue



*Ilustración 16 Diagrama de Despliegue.
Fuente: Autores.*

3.2. DISEÑO ARQUITECTONICO.

Consta de 3 capas en este caso la capa presentación o interfaz que interactúa directamente con el cliente le permite el acceso, registro, modificación y consulta, la capa de la lógica de aplicación en la cual se ejecutan las operaciones para que el sistema ejecute su tarea y la capa de acceso a datos recursos que es la responsable de mostrar todo el sistema de información es decir muestra los datos en este orden al ser una aplicación web se ajusta a este modelo de 3 capas.



Funcionalidad de las Capas

Capa de Presentación o Interfaz de Usuario: Esta capa, está formada por los formularios y los controles que se encuentran en los formularios. Es la capa con la que interactúa el usuario.

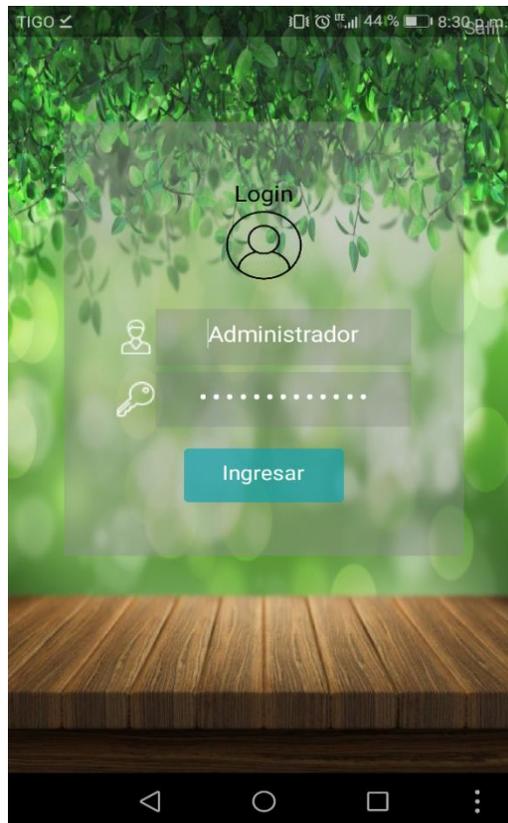
Capa de Negocio: Está formada por las entidades, que representan objetos que van a ser manejados o utilizados por toda la aplicación. En este caso, están representados por clases y “DataTables” que se crean.

Capa de Acceso a Datos: Contiene clases que interactúan con la base de datos, éstas clases altamente especializadas se encuentran en la arquitectura del sistema y permiten, utilizando

los procedimientos almacenados generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio. (www.ecured.cu/Arquitectura_en_Capas)

3.3. DISEÑO INTERFACE

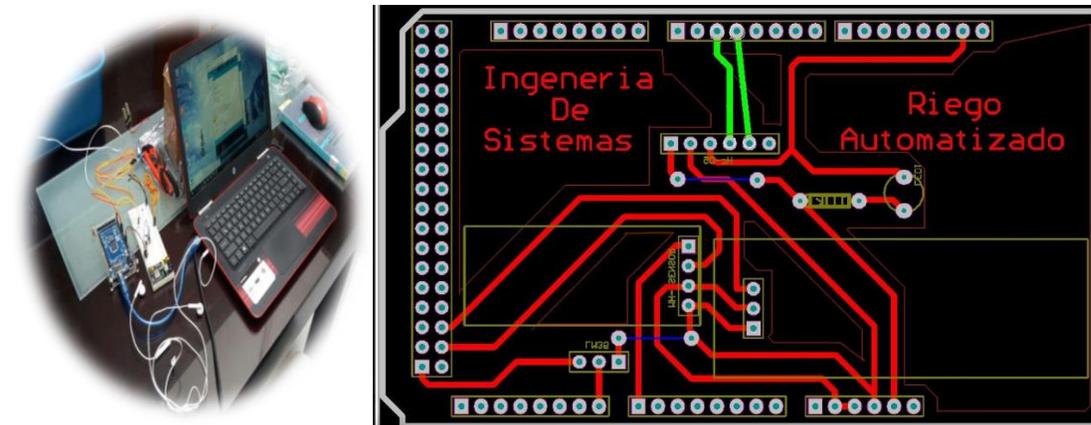
Es una aplicación de interfaz app que en la presentación como control de seguridad permite la validación del usuario o el registro del mismo, como muestra la siguiente ilustración permitiendo así el acceso a esta aplicación móvil.



*Ilustración 17 Interfaz de Usuario Validación.
Fuente: Autores.*

3.4. DISEÑO DEL CIRCUITO

Es un diseño que inicialmente se conectó en una placa de pruebas llamada protoboard con el fin de realizar las pruebas electrónicas pertinentes con los elementos empleados para este montaje, posterior a su funcionamiento se realiza uso de la herramienta Altium Designer.



3.5. DISEÑO DE SEGURIDAD Y CONTROLES

Todo proyecto debe ofrecer seguridad y controles, garantizando de esta manera la Integridad en la información, la cual comprende:

Como se muestra en la ilustración anterior permite una validación de usuario para ingresar a la app, con la autorización de ingreso permite la conexión por bluetooth emparejando el circuito con el dispositivo que tiene instalada la app y que a su vez se encuentra registrada, de este modo brindamos la seguridad en los controles de acceso.

- **De la Base de Datos**

000WebHost Líderes en web hosting gratis es el líder mundial de web hosting gratuito de primera clase, sin publicidad ¡sin ningún coste! Millones de usuarios 000Webhost son nuestra prueba.

Inacabable espacio de disco & ancho de banda

Cada cuenta tiene prácticamente ilimitado espacio y ancho de banda, con incontables conexiones a nuestros servidores. Tendrás las últimas versiones de PHP y MySQL. (<https://co.000webhost.com>)

- **Matriz de Funciones Vs. Usuarios**

FUNCIONES	USUARIO 1	USUARIO 2
Funciones de Acceso e interacción con el sistema	Usuario	Administrador

Tabla 5 Matriz de Funciones vs Usuarios.

- **Roles.**

000WebHost implementa reglas de seguridad para poder definir qué se puede hacer con los datos almacenados en la base de datos, quién puede leer o escribir, entre otras cosas. Para ello cuenta con la opción de crear usuarios por medio del app y por medio de unas reglas de seguridad se adiciona los permisos a los diferentes usuarios.

El Usuario decide, descarga y registra su contraseña para acceder a la herramienta, de esta forma interactúa de manera amigable con el programa realizando consultas de verificación de datos.

- **Permisos**

Los permisos se asignan a la persona interesada en adquirir el producto orientándola a la descarga de la app para su móvil y registrándola en la base de datos para que pueda acceder al sistema.

3.6. SELECCIÓN DE LA HERRAMIENTA DE DESARROLLO

Para la realización de este proyecto se hizo uso de las herramientas de desarrollo PHP y APP Inventor junto lenguaje Arduino que tiene como base de C++.

3.1.6. Codificación

	Descripción	Observación de las pruebas realizadas	cantidad de pruebas
 bd_consultar_datos.php	Archivo que contiene datos de prueba para la realización de gráficas.	Este archivo fue utilizado inicialmente para las pruebas de graficar datos, una vez el resultado en la vista de las gráficas fue satisfactorio se procedió a utilizar el archivo .php () con el cual se consultaban los datos en tiempo real.	1
 bd_insert_user.php	Contiene código para la inserción de nuevos usuarios a la base de datos.	La prueba de inserción fue exitosa desde un inicio lo cual evito que se hicieran cambios a nivel de código	1
 bd_user.php	Consulta todos los usuarios registrados	Se realizaron 2 pruebas con este archivo una consultando solo la información general como nombres y apellidos y otra obteniendo toda la información de los usuarios ya que de esta manera otros módulos como el de modificación podrían acceder más fácil a la información sin necesidad de crear un nuevo archivo para consultas.	2
 conex.php	Contiene la cadena de conexión a la base de datos	La prueba realizada inicial mente dio como resultado un mensaje de : "error en la conexión" ya que los parámetros enviados no eran los correctos, posteriormente se procedió a corregir los errores en los parámetros que se enviaban hasta obtener un mensaje de : "Conexión correcta"	1

 consultarusuarios.php	Código utilizado para verificar que el usuario que desea ingresar al sistema por medio de la aplicación este registrado	Para la verificación del correcto funcionamiento de este archivo se realizaron pruebas con usuarios registrados y no registrados lo que dio un resultado satisfactorio desde un inicio lo que evito que se realizaran cambios posteriores en el archivo	2
 eliminarusuario.php	Elimina un usuario determinado de la base de datos	La prueba de eliminación fue exitosa desde un inicio lo cual evito que se hicieran cambios en el archivo	1
 graficarhumedad.php	Archivo que permite graficar los datos tomados del sensor de humedad Vs. Tiempo	Inicialmente se probó realizar las gráficas haciendo uso de ciclos for para la organización y envío de los datos a google charts pero los resultados obtenidos no fueron satisfactorios, posteriormente se hizo uso de archivos Json lo cual hizo más eficiente el código.	2
 graficarhumedadbarras.php	Archivo que permite graficar en forma de barras los datos tomados del sensor de humedad	Las pruebas realizadas con el archivo graficarhumedad.php ayudaron para que la creación de las demás archivos para graficar se hicieran de una manera similar y así ahorrar tiempo. La prueba del funcionamiento fue exitosa por lo cual no se realizaron cambios en el código.	1
 graficartemperatura.php	Archivo que permite graficar los datos tomados del sensor de temperatura Vs. Tiempo	Las pruebas realizadas con el archivo graficarhumedad.php ayudaron para que la creación de las demás archivos para graficar se hicieran de una manera similar y así ahorrar tiempo. La prueba del funcionamiento fue exitosa por lo cual no se realizaron cambios en el código.	1
 graficartemperaturabarras.php	Archivo que permite graficar en forma de barras los datos tomados del sensor de temperatura	Las pruebas realizadas con el archivo graficarhumedad.php ayudaron para que la creación de las demás archivos para graficar se hicieran de una manera similar y así ahorrar tiempo. La prueba del funcionamiento fue exitosa por lo cual no se realizaron cambios en el código.	1
 insertardatossensor.php	Contiene el código que permite insertar información de los riegos realizados en la base de datos	Inicialmente se probó la inserción con datos de prueba pero la inserción no se realizó de la manera esperada, se procedió a corregir los errores en el código y posterior a esto se probó con datos reales enviados por la aplicación móvil y el resultado esperado fue exitoso.	2

 modificarusuario.php	Código utilizado para modificar la información de los usuarios registrados en el sistema	la primera prueba realizada no dio un resultado exitoso ya que habían errores en el código, se procedió a corregirlo y se probó de nuevo lo que dio un resultado exitoso en la modificación para verificar que se hubiera realizado de manera exitosa se consultó el registro en la base de datos.	2
---	--	--	---

Las líneas de código del Arduino se encuentran en el Anexo I.

4. ANALISIS DE RESULTADOS Y CONCLUSIONES

En esta sección se especifican los diferentes programas del aplicativo y también se relacionan los procesos (Módulos).

Programa	Descripción	Módulo de Proceso Afectado	Tipo Usuario
insert_user.php	Este programa permite registrar un usuario por primera vez	Inserción	Usuario
consultar_datos.php	Permite la búsqueda de datos ya registrados para acceso a la aplicación.	Consulta y Validación	Usuario
Bd_user.php	Permite realizar la consulta en la base de datos con un usuario y contraseña específica para validar el ingreso al sistema.	Login	Usuario
Conex.php	Realiza la conexión a la base de datos	Inserción Validación Consulta Eliminación Modificación	Usuario
Conexión por Bluetooth	Permite conexión al dispositivo para emparejar los dos dispositivos de contacto.	Validación	Usuario
Graficarhumedad.php	Muestra la estadística del comportamiento de la humedad.	Consulta	Administrador

			Usuario
Graficartemperatu ra.php	Muestra el comportamiento del sensor de temperatura con relación al tiempo	Consulta	Administrador Usuario

*Tabla 6 Modulos.
Fuente: Autores.*

4.1. PRUEBAS

4.1.1. Pruebas de Función

Con esta prueba se garantiza que en el ejercicio se realice el ingreso de datos (Entradas), se procesen y se verifique la salida (resultados).

PRUEBAS DE CAJA BLANCA

Tipo: Caja Blanca		
Modulo: inser_user.php		
Entrada	Proceso	Salida
	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">INGRESO DE DATOS</div>	
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Registro de los datos por parte del usuario</div>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Ingresa los datos del usuario</div>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Carga el modulo principal</div>
Se registra un usuario al	Se ingresa el usuario al	Se ejecuta el Modulo

sistema.	sistema y se valida su registro.	de inserción en la Base Datos MySQL.
----------	----------------------------------	--------------------------------------

Tabla 7 Módulo Inser User. Prueba Caja Blanca.

PRUEBAS DE CAJA BLANCA

Tipo: Caja Blanca		
Modulo: consultar_datos.php		
Entrada	Proceso	Salida
<p>Permite la validación</p> <p>Se digita los médicos por parte del asistente del sistema.</p>	<p>Muestra datos insertados</p> <p>Se registran los usuarios en el sistema</p> <p>Se verifica la inserción de usuarios en la Base de Datos MySQL.</p>	<p>Verificación de captura de datos</p> <p>Se consulta el ingreso de usuarios.</p>
<p>Resultados y Observaciones:</p> <p>El proceso evoluciono correctamente y se obtuvieron los resultados esperados por parte del usuario del sistema.</p>		

Tabla 8 Consultar datos. Prueba Caja Blanca.

PRUEBAS DE CAJA BLANCA

Tipo: Caja Blanca		
Modulo: Bd_user.php		
Entrada	Proceso	Salida
<div style="border: 1px solid black; width: 150px; height: 50px; margin: 0 auto; text-align: center; padding: 5px;">Ingreso de datos</div> <div style="border: 1px solid black; width: 150px; height: 50px; margin: 0 auto; text-align: center; padding: 5px;">Ingreso de datos</div> <p>Se digita en la interfaz para garantizar el ingreso a la aplicación</p>	<div style="border: 1px solid black; width: 150px; height: 50px; margin: 0 auto; text-align: center; padding: 5px;">Ingreso de datos</div> <div style="border: 1px solid black; width: 150px; height: 50px; margin: 0 auto; text-align: center; padding: 5px;">Actualización de registros</div> <p>Se ejecuta la opción seleccionada en el Modulo para realizar actualización</p>	<div style="border: 1px solid black; width: 150px; height: 50px; margin: 0 auto; text-align: center; padding: 5px;">Eliminar registros</div> <p>Se realizan las acciones respectivas del Modulo</p>

Tabla 9 BD_User Prueba caja blanca.

PRUEBAS DE CAJA NEGRA

Las condiciones límite son aquellas que se hayan en los márgenes de la clase de equivalencia, tanto de entrada como de salida. Por ello, se ha desarrollado el análisis de valores límite como técnica de prueba. Esta técnica nos lleva a elegir los casos de prueba que ejerciten los valores límite.

- Las pautas que se tuvieron en cuenta para los casos de prueba son:
- Si una condición de entrada especifica un número de valores, se diseñaron dos casos de prueba para los valores mínimo y máximo, además de otros dos casos de prueba para valores justo por encima del máximo y justo por debajo del mínimo.
- Se aplicaron las reglas anteriores a los datos de salida.
- Si la entrada o salida de un programa es un conjunto ordenado, habrá que prestar atención a los elementos primero y último del conjunto.

TIPO	MODULO	PROCEDIMIENTO	RESULTADO
Pruebas para valores limites	Todos	Captura de validación de los rangos permitidos	OK
*OBSERVACIONES			
La captura de validación de los rangos permitidos se efectuó correctamente.			

Tabla 10 Prueba Caja Negra.

4.2.2. Pruebas Modulares

Estas pruebas permitieron verificar la integridad y la operacionalidad de los diferentes módulos del aplicativo.

4.2.3. Pruebas del Sistema

Este tipo de pruebas se efectuó para evaluar el desempeño general de la aplicación y el sistema.

Comprende las siguientes pruebas:

Pruebas de Integración Cada módulo esta en relación con otros, se probaron independientemente y luego se realizó una prueba integral del sistema.

Pruebas de Rendimiento Se verifico la ejecución de cada uno de los programas y el sistema en general, además se realizaron pruebas de rendimiento.

Pruebas de Consistencia Se realizaron las pruebas de consistencia en cada uno de los módulos, durante la ejecución del programa, además se actualizaron cada uno de los módulos del aplicativo.

4.2.4. Prueba de Interfaz

A través de estas pruebas se verificaron las diferentes Interfaces que le permiten al usuario acceder al programa principal y navegar a través de él.

El contenido de la información dentro de las ventanas es accesible adecuadamente con el ratón, flechas de función, flechas de dirección y teclado. Cuenta con barras deslizantes, cuadros de dialogo, botones, iconos y barra de herramientas en algunos módulos propios que así lo requieran

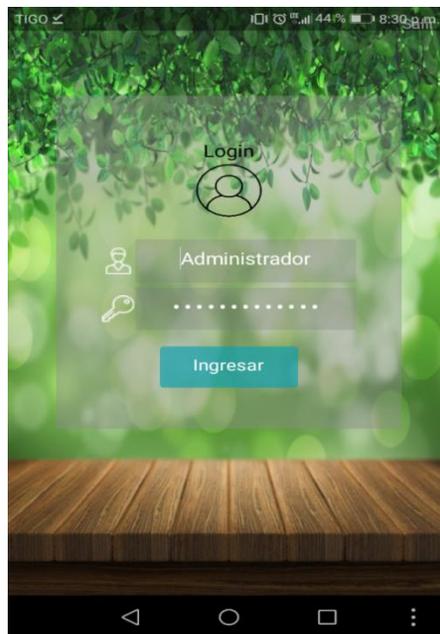


Ilustración 18 Interfaz de Ingreso.



Ilustración 19 Interfaz Menú principal.

4.2.5. Prueba de Calidad

Esta prueba permitió medir factores de un producto de Software, tales como: La usabilidad o facilidad de Uso, la amigabilidad para con el usuario, su entorno gráfico, su nivel de ayuda. Todos estos factores se evaluaron en las secciones anteriores, por lo tanto podemos decir que el producto fue diseñado y desarrollado con estándares de Calidad que garantizan su confiabilidad.



Ilustración 20 Interfaz Sensor de Humedad.

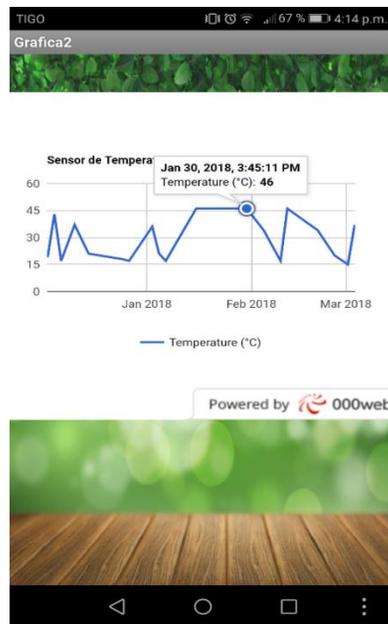


Ilustración 21 Interfaz Sensor de Temperatura.



Ilustración 22 Interfaz Consolidado de Riego.

4.3.INFORME DE PRUEBA

5.	Modulo	Interfaz de Ingreso
	Entrada	. Se realiza el ingreso al sistema, se solicita validación
	Proceso	Se realiza la acción respectiva que tiene como finalidad conexión al sistema de riego.
	Salida	Se ejecuta el Modulo principal del aplicativo.
	Resultado	Se obtuvo un resultado apropiado y aceptado por parte del usuario final.

Tabla 11 Informe Interfaz Usuario.

	Modulo	Interfaz de ingreso
	Entrada	Valida usuario e ingresa y conecta al módulo bluetooth.
	Proceso	Se ejecuta la opción seleccionada en el Modulo.
	Salida	Registro de los datos en la Base de Datos MySQL.
	Resultado	Se obtuvo un resultado apropiado y aceptado por parte del usuario final.

Tabla 12 Informe Interfaz de Ingreso.

	Modulo	Interfaz de conexión
	Entrada	Conecta al dispositivo bluetooth con el celular
	Proceso	Se ejecuta la opción seleccionada en el Modulo.
	Salida	Muestra datos del inicio de del riego, humedad y temperatura
	Resultado	Se obtuvo un resultado apropiado y aceptado por parte del usuario final.

Tabla 13 Informe Interfaz de Conexión

Tipo de pruebas generales	SI	NO
	Cumple	Cumple
Acceso al sistema de acuerdo al perfil y a los parámetros definidos.	X	
Acceso a cada uno de los Módulos que conforman el sistema.	X	
Validación de la información por parte del sistema	X	
Ejecución de cada una de las acciones del sistema.	X	
Navegabilidad dentro del sistema	X	
Acceso a los niveles de ayudas	X	
Pruebas de integración	X	
Pruebas de resistencia	X	
Pruebas de rendimiento	X	
Pruebas de compatibilidad	X	
Pruebas de Usabilidad	X	

Tabla 14 Informe General.

5.3.CONCLUSIONES

Trabajando en los tiempos estipulados y cumpliendo con las tareas asignadas se diseñó e implementación de un Prototipo Sistema Automatizado de Riego para Jardines.

Se diseñó una estructura base para la implementación del prototipo haciendo uso de arduino se implementó un prototipo con sensores y circuitos electrónicos.

Se elaboró una aplicación de software que se integra con arduino para la automatización del sistema de riego.

La Facultad de Ingeniería y Ciencias Básicas en la Misión habla de Formar integralmente profesionales competentes, innovadores, emprendedores y con capacidad de apropiar y liderar procesos tecnológicos, económicos, sociales, que basen su actuar en una sólida fundamentación científica, social y disciplinar, con visión global; por lo tanto en base a esta Misión y teniendo en cuenta el proyecto ejecutado y presentado, cumplimos con las competencias básicas para ser profesionales íntegros y capaces de apoyar y ejecutar proyectos de alcance, mostrando así una aplicación con calidad que garantiza operativamente, los requerimientos de los clientes.

5.4. RECOMENDACIONES

Se recomienda que se realice una actualización periódica del sistema.

Se recomienda a futuro optimizar la conexión de bluetooth a conexión inalámbrica para que de esta forma tenga menos limitante en su conexión.

Se recomienda dejar un módulo en el que el usuario pueda cambiar el parámetro del sensor de humedad en caso que conozca las características de las plantas, haciendo uso responsable para evitar el daño a las mismas.

BIBLIOGRAFIA

- Barón Alarcón, J. E. (2017). Obtenido de <http://repository.libertadores.edu.co/handle/11371/1292>
- Ferrepat.* (s.f.). Obtenido de <http://www.revista.ferrepat.com>
- Hostinet.* (2016). Obtenido de <https://www.hostinet.com/formacion/panel-alojamiento>
- http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/SC/WWDR2015Facts_Figures_SPA_web.pdf. (s.f.).
- <https://co.000webhost.com>. (s.f.). <https://co.000webhost.com>. Obtenido de <https://co.000webhost.com>
- <https://definicion.de/base-de-datos/>. (s.f.). Obtenido de <https://definicion.de/base-de-datos/>
- <https://openwebinars.net>. (2014). <https://openwebinars.net>. Obtenido de <https://openwebinars.net>
- <https://www.redeszone.net>. (s.f.). <https://www.redeszone.net>.
- <https://www.softeng.es>. (s.f.). <https://www.softeng.es>.
- Libertadores, F. U. (s.f.). <http://www.ulibertadores.edu.co/facultad-ingenieria/>
- mapama.* (2017). Obtenido de <http://www.mapama.gob.es>
- Normatividad_Derechos_de_Autor. (s.f.). *Cide.* Obtenido de http://www.cide.edu.co/cidevirtual/file.php/1/Normatividad_Derechos_de_Autor.pdf
- ORACLE.* (s.f.). Obtenido de <https://www.oracle.com/co/mysql/index.html>
- Reloj.* (2017). Obtenido de https://es.wikipedia.org/wiki/Reloj_en_tiempo_real
- SCRUM.* (2014). Obtenido de <http://www.i2btech.com/blog-i2b/tech-deployment/para-que-sirve-el-scrum-en-la-metogologia-agi>
- Softeng.* (2015). Obtenido de <https://www.softeng.es>
- Traxco.* (2010). Obtenido de <https://www.traxco.es/blog/noticias-agricolas/historia-ancestral-del-riego>
- WRG. (2009). Obtenido de <https://www.2030wrg.org/login-required/>
- WWAP, H. (2015). Obtenido de <http://www.unesco.org/new/es/natural-sciences/environment/water/wwap/wwdr/2015-water-for-a-sustainable-world/#c1485604>
- www.ecured.cu/Arquitectura_en_Capas. (s.f.). Obtenido de www.ecured.cu/Arquitectura_en_Capas
- www.unesco.org. (2015). <http://www.unesco.org/new/fileadmin>.

ANEXOS

Sprint del punto 3.2. Metodología del desarrollo del sistema. Sprint backlog organizados de la siguiente manera.

Anexo A

Sprint 1

Planeación

Tareas Pendientes	0
Total Horas Estimadas	25
Total Horas Ejecutadas	34

4-Sept	5-Sept	6-Sept	7-Sept	8-Sept	9-Sept	11-Sept	12-Sept	13-Sept
--------	--------	--------	--------	--------	--------	---------	---------	---------

Pila Sprint

Tarea	Tipo	Estado	Responsable	Estimación Inicial	Esfuerzo								Esfuerzo ejecutado por Tarea	
Análisis de Requerimientos	Análisis	Terminado	Diana Cañon	8	6	2								8
Bosquejo	Análisis	Terminado	Edna Cifuentes	5		3	2	2	2					9
Diseño del Prototipo	Análisis	Terminado	Diana y Edna	7				2	3	2	2	2		11
Materiales Prototipo	Compra	Terminado	Diana y Edna	5			4	2						6
Total Horas Trabajo Ejecutado				25	6	5	2	6	6	3	2	2	2	34

Anexo B**Sprint 2** Documentación e Investigación

Tareas Pendientes	0
Total Horas Estimadas	20
Total Horas Ejecutadas	28

14-Sept	15-Sept	16-Sept	18-Sept	19-Sept	20-Sept	21-Sept	22-Sept	23-Sept
---------	---------	---------	---------	---------	---------	---------	---------	---------

Pila Sprint

Tarea	Tipo	Estado	Responsable	Estimación Inicial	Esfuerzo										Esfuerzo ejecutado por Tarea	
Introducción	Análisis	Terminado	Diana Cañon	1	1	1										2
Descripción del Problema	Análisis	Terminado	Edna Cifuentes	5	1	2	1	1								5
Justificación	Análisis	Terminado	Diana y Edna	3			1	1	2							4
Objetivos	Análisis	Terminado	Diana y Edna	2	1	1										2
Antecedentes Históricos	Consulta	Terminado	Edna Cifuentes	4					2	2	2					6
Descripción de Materiales	Prototipo	Terminado	Diana Cañon	2			2	1								3
Marco legal	Consulta	Terminado	Diana Cañon	3							2	2	2			6
Total Horas Trabajo Ejecutado				20	3	4	2	2	4	2	4	2	2			28

Anexo C

Sprint 3 Diseño Estructural

Tareas Pendientes	0
Total Horas Estimadas	82
Total Horas Ejecutadas	80

25-Sept	26-Sept	27-Sept	28-Sept	29-Sept	30-Sept	1-Oct	2-Oct	3-Oct
---------	---------	---------	---------	---------	---------	-------	-------	-------

Pila Sprint

Tarea	Tipo	Estado	Responsable	Estimación Inicial	Esfuerzo										Esfuerzo ejecutado por Tarea
Diseño del Circuito	Análisis	Terminada	Diana Cañon	26	4	4	1	1	4	2		6	4	26	
Diagramas UML	Análisis	Terminada	Diana Cañon	12	2		2		2	4		2		12	
Diseño de Interfaz	Análisis	Terminada	Edna Cifuentes	30	6	2	2	6	6	2		2	2	28	
Definición de herramientas de desarrollo	Análisis	Terminada	Edna Cifuentes	4	1	2	1							4	
Creación de Base de Datos	Análisis	Terminada	Edna Cifuentes	10	2	2	2	2					2	10	
Total Horas Trabajo Ejecutado				82	15	10	6	7	12	8	0	10	8	80	

Anexo D**Sprint 4****Codificación**

Tareas Pendientes	2
Total Horas Estimadas	47
Total Horas Ejecutadas	47

12-Oct	13-Oct	14-Oct	17-Oct	18-Oct	19-Oct	20-Oct	21-Oct	23-Oct
--------	--------	--------	--------	--------	--------	--------	--------	--------

Pila Sprint

Tarea	Tipo	Estado	Responsable	Estimación Inicial	Esfuerzo								Esfuerzo ejecutado por Tarea	
Conexión Interfaz	Análisis	Terminada	Edna y Diana	7							2	3	2	7
Codificación Arduino	Análisis	Terminada	Diana Cañon	10	2	2	6	2						12
Codificación Interfaz	Análisis	Terminada	Edna Cifuentes	30	8	2	2	2	8	2		2	2	28
Codificación Programación en Bloques	Análisis	Terminada	Edna Cifuentes	20	3	5	5	4				4		21
Total Horas Trabajo Ejecutado				47	10	4	8	4	8	2	2	5	4	47

Anexo E

Sprint 5

Ajustes

Tareas Pendientes	0
Total Horas Estimadas	36
Total Horas Ejecutadas	33

1-Nov	2-Nov	3-Nov	4-Nov	7-Nov	8-Nov	9-Nov	10-Nov	11-Nov
-------	-------	-------	-------	-------	-------	-------	--------	--------

Pila Sprint

Tarea	Tipo	Estado	Responsable	Estimación Inicial	Esfuerzo										Esfuerzo ejecutado por Tarea
Mejora del Prototipo	Diseño	Terminada	Edna Cifuentes	16	2	2	1	3	1	1	1	4	1	16	
Mejoras del circuito	Diseño	Terminada	Diana Cañon	20	2	2		2		3		2	6	17	
Total Horas Trabajo Ejecutado				36	4	4	1	5	1	4	1	6	7	33	

Anexo F

Sprint 6 Pruebas

Tareas Pendientes	0
Total Horas Estimadas	24
Total Horas Ejecutadas	27

14-Nov	15-Nov	16-Nov	17-Nov	18-Nov	20-Nov	21-Nov	22-Nov	23-Nov
--------	--------	--------	--------	--------	--------	--------	--------	--------

Pila Sprint

Tarea	Tipo	Estado	Responsable	Estimación Inicial	Esfuerzo										Esfuerzo ejecutado por Tarea
Pruebas del circuito	Diseño	Terminada	Diana Cañon	12	2	2	2	2	2	2					12
Prueba de Conexión Interfaz	Prueba	Terminada	Edna y Diana	6									3	3	6
Prueba del Sistema de Riego	Prueba	Terminada	Edna y Diana	3								2	3		5
Prueba Arduino	Análisis	Terminada	Diana y Edna	3							4				4
Total Horas Trabajo Ejecutado				24	2	2	2	2	2	2	0	3	3		27

Anexo G

Sprint 7 Verificación

Tareas Pendientes	0
Total Horas Estimadas	10
Total Horas Ejecutadas	11

28-Nov	29-Nov	30-Nov	1-Dic	2-Dic	4-Dic	5-Dic	6-Dic	7-Dic
--------	--------	--------	-------	-------	-------	-------	-------	-------

Pila Sprint

Tarea	Tipo	Estado	Responsable	Estimación Inicial	Esfuerzo										Esfuerzo ejecutado por Tarea	
Módulo de Ingreso	Prueba	Terminada	Edna y Diana	2	1	1										2
Validación de Usuario	Prueba	Terminada	Edna y Diana	2		1	1									2
Módulo de Conexión	Prueba	Terminada	Edna y Diana	2			1	1	1							3
Registro de Usuarios	Prueba	Terminada	Edna y Diana	2					1						1	2
Actualización de Usuarios	Prueba	Terminada	Edna y Diana	2							1			1		2
Total Horas Trabajo Ejecutado				10	1	2	2	1	2	0	1	0	2			11

Anexo H

Sprint 8

Documentación
Final

Tareas Pendientes	0
Total Horas Estimadas	47
Total Horas Ejecutadas	47

10-Ene	11-Ene	12-Ene	13-Ene	16-Ene	17-Ene	18-Ene	19-Ene	20-Ene
--------	--------	--------	--------	--------	--------	--------	--------	--------

Pila Sprint

Tarea	Tipo	Estado	Responsable	Estimación Inicial	Esfuerzo										Esfuerzo ejecutado por Tarea
Elaboración Manual Técnico	Documentación	Terminada	Edna Cifuentes	10	2	1	1		1	1	1	1	2	10	
Elaboración Manual de Usuario	Documentación	Terminada	Diana Cañon	10	2	2	2	2	2					10	
Revisión del Documento	Asesorías	Terminada	Edna y Diana	15	2	2		4		2		4	1	15	
Resultado de Graficas	Prueba	Terminada	Edna y Diana	6			2		2		2			6	
Revisión de Graficas	Prueba	Terminada	Edna y Diana	6			2		2		2			6	
Total Horas Trabajo Ejecutado				47	6	5	7	6	7	3	5	5	3	47	

Anexo I

Código Arduino

```

#include <SoftwareSerial.h>
#include <Wire.h>
#define relay 50
#define senshumedad A7
#define senstemp A8

SoftwareSerial BT(2,3); // RX, TX recorder que se cruzan

uint8_t second, minute, hour, wday, day, month, year, ctrl;
int valor, temp, valorhumedad;
float voltaje, grados;

void setup()
{
  Serial.begin(9600);
  Wire.begin();
  BT.begin(9600);
  pinMode(relay, OUTPUT);
  pinMode(senshumedad, INPUT);
  pinMode(senstemp, INPUT);
}

void read_datos()
{
  temp = analogRead(senstemp);
  voltaje = (temp / 1023.0) * 5000;
  grados = voltaje / 10;

  valorhumedad = analogRead(senshumedad);
  valorhumedad = map(analogRead(senshumedad), 0, 1023, 100, 0);
}

void enviar_datos()
{
  BT.print(valorhumedad);
  BT.print('|');
  BT.print(grados);
  BT.print('|');
}

```

```
BT.print(year);
BT.print('-');
BT.print(month);
BT.print('-');
BT.print(day);
BT.print('|');
BT.print(hour);
BT.print(':');
BT.print(minute);
BT.print(':');
BT.print(second);
BT.println();

delay(1000);
}

void print_datos()
{
  Serial.print("Fecha: ");
  Serial.print(day);
  Serial.print('/');
  Serial.print(month);
  Serial.print('/');
  Serial.print(year);

  Serial.print(" Hora: ");
  Serial.print(hour);
  Serial.print(':');
  Serial.print(minute);
  Serial.print(':');
  Serial.print(second);

  Serial.print(" Temperatura: ");
  Serial.print(temperatura);
  Serial.println();

  Serial.print(" Humedad: ");
  Serial.print(valorhumedad);
  Serial.println();
}

bool read_reloj()
{
  Wire.beginTransaction(0x68);
  Wire.write(0x00);
  if (Wire.endTransmission() != 0)
    return false;
}
```

```

Wire.requestFrom(0x68, 8);
second = bcd2bin(Wire.read());
minute = bcd2bin(Wire.read());
hour = bcd2bin(Wire.read());
wday = bcd2bin(Wire.read());
day = bcd2bin(Wire.read());
month = bcd2bin(Wire.read());
year = bcd2bin(Wire.read());
ctrl = Wire.read();
return true;
}

uint8_t bcd2bin(uint8_t bcd)
{
    return (bcd / 16 * 10) + (bcd % 16);
}

void loop()
{
    int valorhumedad = analogRead(senshumedad);
    valorhumedad = map(analogRead(senshumedad), 0, 1023, 100, 0);

    while (valorhumedad > 20)
    {
        digitalWrite(relay, HIGH);
        valorhumedad = analogRead(senshumedad);
        valorhumedad = map(analogRead(senshumedad), 0, 1023, 100, 0);
    }

    if (valorhumedad < 20)
    {
        digitalWrite(relay, LOW);
        if (read_reloj())
        {
            read_datos();
            print_datos();
            enviar_datos();

            valorhumedad = analogRead(senshumedad);
            valorhumedad = map(analogRead(senshumedad), 0, 1023, 100, 0);

            while (valorhumedad < 20)
            {
                valorhumedad = analogRead(senshumedad);
                valorhumedad = map(analogRead(senshumedad), 0, 1023, 100, 0);
            }
        }
    }
}

```

```
}  
delay(1000);  
}
```