

Resumen Ejecutivo.

En el presente reporte, se muestran a detalle la planificación y elaboración de un sistema de control automatizado para el riego de plantas. Esto con la finalidad de poder satisfacer la necesidad principal en el área de aplicación que concierne al cuidado y mantenimiento de un jardín, donde algo muy importante y fundamental es lograr aumentar la vida de las plantas y concientizar sobre el cuidado del agua, al igual que dar un interesante inicio al desarrollo de prototipos implementados en casas inteligentes.

Así entonces, el proceso constó en lograr crear un sistema con un cierto grado considerable de automatización para riego de plantas de jardín, un proyecto que tuvo un enfoque especial dentro del hogar, a lo que se puede considerar incluso como una aplicación a la moderna domótica^{cita}. El diseño y desarrollo del proyecto se dividieron en dos fases generales, la primera se enfocó en la creación de una aplicación web por medio de la cual se dio seguimiento a los análisis del sistema desde cualquier dispositivo. La segunda etapa tuvo su enfoque en el diseño y ensamblaje del sistema, el cual tuvo como componentes un sensor de humedad de suelo y una bomba de agua, considerada esta última como el elemento alternativo para el cuidado del agua.

Puede verse a una tercera fase como la de aplicación para pruebas y fallo, antes de su presentación final, el producto mínimo viable fue sometido a una rigurosa fase de pruebas para comprobar su efectividad y gran calidad, ya que si se desea dar continuidad a la creación de dispositivos inteligentes, es de vital importancia cerciorarse de que estos tengan un excelente funcionamiento y puedan brindar un servicio de gran nivel, y así poder dar continuidad a nuevos diseños, que algunos quizá, sean desarrollados con mejores y más modernas tecnologías.

El modelo sistemático del sistema no difiere de un modelo sistemático convencional, constó de sensores para captar múltiples señales, una fase de control y actuadores. Para el buen funcionamiento de la parte de control, se contó también con la utilización de un microcontrolador ATMEGA328P-PU, el cual es completamente compatible con estos dispositivos. Posterior a la planificación y ejecución de las debidas investigaciones, experimentaciones y desde luego el realizado del producto mínimo viable, se espera que futuras generaciones den continuidad a esta investigación dentro de esta misma área que bien puede desviarse hacia la industria agrícola, que siempre estará demandando innovación.

Introducción

El proyecto denominado "Diseño de Software para Sistema de Riego" fue desarrollado para la asignatura de Estadística con la colaboración del Ingeniero Gilberto Sánchez Mares como asesor de proyecto y el alumno de décimo cuatrimestre de la carrera de Ingeniería en Electrónica, Agustín Romo Lara como practicante del proyecto. El proyecto tiene como objetivo solucionar una problemática que busca la automatización en el tratado y cuidado de plantas de jardín, es por ello que se acude a la investigación y conocimiento de nuevas herramientas de diseño y desarrollo para buscar una solución autómata, confiable y viable para el diseño del proyecto.

Dicho proyecto es muestra clara de las necesidades actuales de cualquier campo enfocado en el cuidado del medio ambiente, implementar soluciones confiables y viables para resolver sus problemáticas, las cuales son atacadas con el desarrollo e implementación de herramientas de software, optimizando sus procesos y de igual forma facilitando su manera de ejecución, es decir, que cualquier usuario pueda manipular el sistema de forma eficaz, y así, poder garantizar el crecimiento y durabilidad de vida de sus plantas. Es por ello que en el presente proyecto se considera necesaria la capacitación en herramientas de diseño y programación tales como Javascript, Node.js con express y la tecnología manejable de Arduino, herramientas que fueron explicadas a lo largo del presente reporte.

Actualmente la implementación de sistemas de riego automatizados en el sector agrícola ha ido incrementando de forma significativa a través de los últimos años, esto gracias al impacto notorio en las finanzas de las empresas de alimentos, por ejemplo, optimizando su margen de ganancias al poder pagar la inversión realizada en los sistemas de riego automatizados en un plazo de tiempo relativamente corto.

El sistema de riego para plantas de jardín contará también con la tarea de utilizar el agua de manera adecuada, esto con la finalidad de evitar desperdiciar el elemento más importante de todo el sistema. La concientización sobre el cuidado del agua en casos como el que se presenta, es un factor que no debe pasarse por alto, ya que es el agua la principal fuente que dará vida y durabilidad a las plantas y al jardín en un contexto general. Por ello, es fundamental realizar un control completo del gasto de agua necesario para evitar el desperdicio de un elemento irrenovable.

Es importante iniciar con un estudio de las tecnologías existentes, para poder evaluar la mejor estrategia de desarrollo del prototipo, y que así sea de gran utilidad para la realización de un sistema de riego de plantas que bien puede ser utilizado como prototipo en campos de la industria agrícola o bien, ser implementado en el hogar. Este proyecto se define como un prototipo a nivel laboratorio que incorporará el diseño e implementación de toda la circuitería electrónica en tablillas de experimentación (protoboard), para pruebas de funcionamiento de los componentes del circuito. Así como realizar la programación, con sus respectivas pruebas de funcionamiento.

El desarrollo del proyecto se dividió en 2 fases de diseño, una de ellas fue diseñar la aplicación web para el seguimiento inalámbrico de los resultados del sistema, donde fue necesaria la capacitación para el conocimiento de las distintas herramientas de diseño y programación a utilizar para la elaboración del software de este prototipo. En un contexto específico, se contará con la implementación del lenguaje de programación Javascript, para la parte del diseño, se implementará un servidor web, Node.js que funciona como la herramienta a cargo correr el código de programación desde el servidor, e irá de la mano con una librería para la creación de aplicaciones web conocida como 'express', todo esto con la finalidad de tener una interfaz web que controlará dispositivos.

La otra fase se centró en el diseño físico y del circuito electrónico que fue el encargado de controlar el funcionamiento del sistema, dicho circuito realizó la parte del riego de agua, este circuito fue probado en tablilla de pruebas (protoboard) con lo cual se pretenden realizar las pruebas necesarias hasta obtener el funcionamiento óptimo del circuito, mismo que fue el proyecto del sistema de control del prototipo en forma. Este circuito una vez impreso y soldado en su PCB, contendrá en su implementación un sensor detector de humedad y una bomba para arrojar el agua, que mueve 2 litros por minuto con un funcionamiento de 6v a 0.3A.

Es importante mencionar que en la parte de diseño electrónico, fue necesario programar un microcontrolador el cual se encargó de comandar la tarjeta electrónica de mando de forma digital para futuras revisiones o modificaciones en el sistema. La comunicación entre el dispositivo de programación y el servidor se dió con la lectura de datos en el puerto USB, medio del cual se dieron las especificaciones de su funcionamiento en el desarrollo del reporte.

Finalmente cuando se hayan realizado las pruebas necesarias, las 2 etapas ya finalizadas se acoplan de tal forma que logren su funcionamiento óptimo entre ellas.

El sistema descrito trata de un prototipo o producto mínimo viable, el cual permita hacer las pruebas verdaderas en el establecimiento donde se requiere y en un futuro realizar el modelo final con las mejoras implementadas.

Capítulo 1.

Generalidades.

Para la materia de Estadía de décimo cuatrimestre de la carrera de Ingeniería Electrónica, de la Universidad Politécnica de Aguascalientes, se ha planeado el desarrollo y diseño de un sistema de riego dentro de las especificaciones y funciones mencionadas, el proyecto dio enfoque a la tarea de proporcionar el buen mantenimiento para plantas de jardín, con la finalidad de prolongar el tiempo de vida de las plantas y concientizar sobre el cuidado del agua durante la realización de esta labor.

1.1 Historia del Proyecto

El origen del problema trata de implementar un sistema automatizado que pueda realizar la tarea de regar plantas de jardín, sin contar con un diseño bastante burdo para así reducir gastos y tiempo en fases de mantenimiento, si bien, ya han sido creados estos tipos de sistemas, este contará con la novedosa implementación de ser controlado por una aplicación web desde el servidor en cualquier navegador.

Poder brindar el mantenimiento adecuado a las plantas es una tarea diaria, además es importante tener en cuenta que no todas plantas requieren del mismo cuidado y mantenimiento, algunas por ejemplo, no requieren ser regadas a diario, otras por lo tanto necesitan de un riego constante y en este caso es importante analizar cuanta es la cantidad necesaria para así también reducir el uso excesivo del agua.

1.2 Actividades

Como primera actividad del cronograma, una vez definida la naturaleza del proyecto, es comenzar con la fase de capacitación, es aquí donde por medio de videos e instrucciones de instalación de diversos softwares y frameworks^{cita} se ha llevado a cabo toda concientización para el correcto aprendizaje en el lenguaje de programación que fue utilizado y las diversas herramientas para el diseño del servidor web, utilizando una plataforma cuyas múltiples clases en vivo brindaron el conocimiento requerido para llevar a cabo el proyecto de esta magnitud de manera exitosa, el

apoyo en esta plataforma de internet, impuso retos de rápido aprendizaje, sus temas se desglosaron en subtemas y ejercicios que ayudaron a facilitar el entendimiento para el lenguaje que está enseñando y así, ser capaz de disipar todas las dudas, de cualquier modo, en caso de que se sigan generando dudas o se presenten problemas en las fases de instalación, el asesor brindará el apoyo necesario basándose en sus conocimientos para despejar las dudas y eliminar esos problemas.

Después de llevar un proceso de capacitación en el lenguaje de Javascript con HTML, y después de haber realizado ejercicios de prueba para el correcto dominio absoluto de los temas cubiertos durante estas presentaciones, se procede a una fase de capacitación más compleja donde lo primero que hay que entender es la diferencia entre un servidor y un cliente, el cliente es el navegador al cual se le abre un archivo HTML, todo lo realizado en la primera fase de capacitación se llevó a cabo en un cliente, entonces desde un archivo HTML se invocaba un archivo de Javascript <script src= "programa.js"></script>. Lo cual ha generado un efecto dentro de su respectiva programación, pero hoy en día se tiene el navegador y todo aquello que se realiza en el navegador se le conoce como Frontend^{cita}. Posterior a esto último se tiene el ya mencionado servidor, cuyo concepto es todo aquello que corre o se ejecuta en internet, donde se encuentran otros dos fundamentales elementos como el Backend^{cita} y la base de datos.

Tanto Frontend como Backend, son los espacios en los que se desarrolla el programador como tal. Actualmente la mayoría de los programadores y desarrolladores suelen centrarse en uno solo, pero ¿cuáles son realmente las diferencias entre Frontend y Backend? Backend es el código de programación que corren los servidores que “viven” en internet y que por lo tanto procesan los datos, por ende Backend es el que posteriormente se conecta con la base de datos. Hay diversos lenguajes de programación en Backend como PHP, Python y Node.js. Node.js es un servidor muy particular cuya tecnología permite correr Javascript en el servidor, es decir, Node.js permitirá correr un código de Javascript en el lado del servidor y ya no desde el lado del cliente en Frontend con HTML.

Es así como la segunda fase de la capacitación se centró en la realización de pruebas y ejercicios para correr el Javascript desde el lado del servidor, cosa que fue fundamental en el desarrollo de la aplicación web. Vease al servidor como el software con el que se ejecutará el Backend, así como en el lado del cliente se utilizó un HTML, del lado del servidor se necesita este software, y es aquí donde aparece Node.js.

La tecnología de Frontend es aquella que funciona en el navegador, sus tres tecnologías fundamentales son HTML, CSS y Javascript. Ahora la manera en cómo se lleva a cabo la utilización de Frontend y Backend es mediante un Stack^{cita}. El Stack hace referencia a todo un grupo o colección de tecnologías para el diseño web que consta principalmente en el sistema operativo sobre el cual trabaja, el software de su

servidor Backend, su base de datos y su lenguaje de programación. Un Stack funciona mediante un conjunto de ramas que se desarrolla paso a paso.

Empezando en el navegador donde se escribe la dirección que termina en .com, al presionar Enter, esa página solicitada va a un servidor especial donde la página solicitada es cambiada por una dirección IP, a todo esto se le conoce como un servidor DNS que cambia el nombre de la página solicitada hacia una IP, después de esto, cuando se tiene la IP el software del servidor web recibe esa información, entonces el lenguaje de programación del servidor se encarga de recibir los datos para ser procesados, dicho lenguaje se conecta a la base de datos, la cual crea un archivo resultante(HTML, CSS y javascript) que son enviadas al navegador, entonces todo este ciclo genera el cliente.

Tras todo lo mencionado, el diseño del proyecto constará en su propio diagrama de tipo Stack, donde desde el navegador se accederá a un servidor local, cuando se corre un servidor local en casa por ejemplo, la IP es 127.0.0.1, una IP que hace referencia a la propia computadora donde se está trabajando. Se correrá un servidor de Node.js, que correrá el código Javascript y posteriormente con el uso de librerías especiales enviará esa información a una tarjeta electrónica Arduino del lado del servidor, ya que es solo por medio de este que se puede establecer la comunicación con dispositivos externos.

En un contexto resumido, el Stack de este proyecto del Sistema de Riego para plantas de jardín, constará de Node.js para correr Javascript del lado del servidor. Express, que es una librería del propio Node.js para crear aplicaciones web. Y Javascript, del lado del cliente en el navegador. Es importante hacer énfasis en la parte de que la razón por la que se tiene que usar Node.js, es porque HTML no puede acceder a dispositivos, mientras que un servidor es como un programa nativo dentro del sistema operativo que tiene acceso a todo, incluso puertos USB.

1.3 Justificación

Actualmente existen diversos sistemas de riego para campos y plantaciones que están basados en subsistemas los cuales constan en tuberías conectadas a las llaves de agua que son activadas de manera automática, cada cierto tiempo o realizando una medida de la temperatura ambiente con la finalidad de detectar si las probabilidades de lluvia son mínimas o máximas. Estos sistemas inteligentes han ido proporcionando mejoras en sus campos de prueba e implementación y son una medida ejemplar para los múltiples proyectos sobre sistemas o robots de riego que se pretenden crear e innovar a lo largo del mundo en los múltiples campos de riego, ya sea de clase industrial agrícola o en el hogar.

El presente proyecto está enfocando su diseño en mejorar la forma de regar plantas de jardín y el propio jardín tomando como justificación la actual problemática existente y la forma en como se trata esa problemática. En una casa es muy común utilizar aspersores para el riego de un jardín e incluso los propios jardineros incluyen en sus actividades proporcionar el riego de forma manual.

En el hogar, la utilización de un aspersor es considerada una medida bastante excesiva, ya que los aspersores tienen un enfoque de desarrollo de rango industrial, para ser usados en cuestiones de agricultura y jardinería en campos extensos, pero pensado en un hogar, si bien un aspersor es muy eficiente y capaz, puede tender a la mayor problemática de todas, el desperdicio de agua. Una de las principales problemáticas a resolver de este proyecto va enfocada al ahorro constante de agua y concientizar al usuario sobre el tipo de plantas y jardín donde planea poner a trabajar al sistema para tener en cuenta cuanta es la cantidad real de agua al día que sus plantas necesitan y de este modo obtener un gran resultado tanto en el cuidado de sus plantas como el cuidado del agua.

Titanium Solutions es una empresa de vanguardia dedicada al diseño y la buena calidad del software. Entre las múltiples habilidades de la creciente compañía se encuentra la implementación del SCRUM. SCRUM es un marco de desarrollo ágil que se basa en adquisición de una manera nueva para el desarrollo sin la necesidad de estructurar un plan completo para el diseño del producto.

Scrum basa sus actividades en tres pasos únicos para testificar el buen avance de los proyectos a la brevedad: ¿Qué hiciste ayer? ¿Qué harás hoy? Y ¿Existe algún problema que impida realizar tus objetivos de hoy? De esta manera, la herramienta SCRUM asegurará una buena calidad dentro del diseño de software.

1.4Objetivo general

Implementar un sistema automático y funcional para el mantenimiento de plantas de jardín en el hogar. La implementación del sistema es un ejemplo para la investigación y el desarrollo en el campo de sistemas para casas inteligentes.

1.4.1Objetivos específicos

Crear un sistema automático de riego de plantas casero, con la capacidad de tomar la decisión de efectuar la tarea de riego sin intervención humana. Las acciones del sistema son visualizadas por medio de una aplicación web en tiempo real, la cual registra todos los datos obtenidos por el sistema y da aviso de cuando este se encuentra efectuando la labor de riego.

1.5 Alcances y limitaciones

Gracias a las tecnologías ya existentes como diversos tipos de sensores capaces de realizar medidas con cierto grado de exactitud y las tecnologías para el diseño de software, es posible realizar un sistema que mide los parámetros necesarios para el mantenimiento de plantas de jardín y la creación de una aplicación web donde pueden ser visualizados. La comunicación serial hizo posible que diversos actuadores trabajaran de forma adecuada y sin fallas. Todo esto logró la creación y manipulación de un sistema de riego automático, que aún así, está muy limitado en cuanto a tecnología inalámbrica se refiere. Todo el circuito maestro del sistema debe permanecer conectado al puerto USB donde se instaló Node.js, esto también hace imposible que mediante la aplicación se manipule el sistema de forma remota, y por ende lo limita a sólo visualizar por medio de la aplicación los estados de los sensores y actuadores.

Capítulo 2.

Diseño y Desarrollo.

A continuación se revelan las actividades llevadas a cabo para efectuar el presente proyecto, enfocadas principalmente al desarrollo del prototipo, la adquisición de los componentes necesarios para su automatización y el proceso en el diseño del software para la aplicación web. El sistema de control da un enfoque global que consta de tres etapas, la captura de señales, la interpretación y digitalización de los datos analógicos y la interacción por medio de actuadores.

Para llevar a cabo este proyecto, se definieron los sensores y actuadores utilizados, entre los que se destaca el sensor medidor de humedad en la tierra, sensor de temperatura ambiente, bomba de agua y servomotor, añadiendo cuestiones de ensamble para la forma del diseño y el circuito principal o tarjeta madre que logre establecer el control correcto para el sistema de riego automatizado. Las tareas fundamentales del sistema, en un plan definido, fueron medir la humedad en la tierra y temperatura ambiente para así realizar la tarea de efectuar el riego, manteniendo un control enfático en el cuidado del agua al momento de realizar esta labor.

Es tarea vital del usuario llevar a cabo una investigación previa del tipo de plantas sobre las que desea trabajar, para asegurarse del tipo de mantenimiento que cada planta necesita.

2.1 Procedimiento

A continuación se muestran los materiales, procedimientos, esquemas de diseño y algunos algoritmos, para la automatización del sistema, utilizado para lograr obtener resultados positivos al momento de realizar la tarea diaria del cuidado de plantas de jardín en el hogar.

Las actividades para el diseño quedaron establecidas en base al siguiente cronograma:

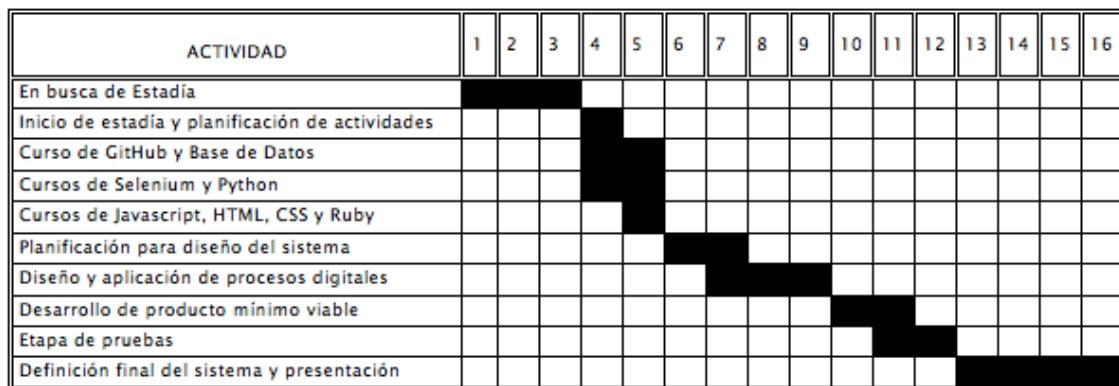


Figura.1. Cronograma de Actividades.

Se puede apreciar que el fundamento en todas las actividades, se basa en la capacitación y el conocimiento de las múltiples herramientas para el diseño de software.

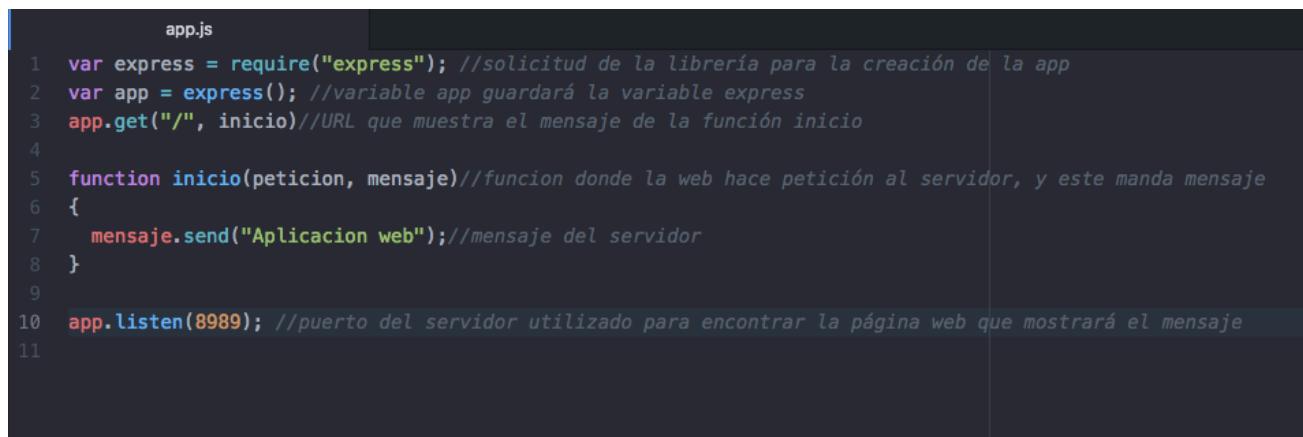
Posterior al conocimiento básico necesario es muy importante realizar pruebas para obtener un dominio suficiente en el conocimiento de estas herramientas, y así poder establecer una planificación absoluta para el diseño y desarrollo del sistema deseado, entonces la actividad posterior a los cursos de capacitación, fue tener una idea clara de lo que se pretende hacer y que aplicaciones fueron utilizadas para lograr los objetivos.

Todas las fases correspondientes a la actividad de planificación para el desarrollo del sistema fueron tratadas a continuación. Al tener en claro que lo necesario para el desarrollo de la aplicación web es hacer todo un diagrama de tipo Stack, se han definido las herramientas para diseño necesarias: El servidor de Node.js, el lenguaje de programación JavaScript, HTML con CSS para la interfaz gráfica, el framework Socket.io para la conexión entre el servidor y la interfaz y la librería 'express'.

Ya que se definió el producto a realizar y las herramientas para llevarlo a cabo, el paso siguiente es para las descargas e instalaciones de todos los frameworks necesarios para llevar a cabo esta actividad. Tras la instalación de Node.js y un Software editor de código para la programación en JavaScript, se efectuará el primer ensayo basado en la prueba del servidor.

La prueba del servidor consta en que mediante un código de Js el usuario pueda ver un mensaje creado por él mismo en una dirección URL que hace referencia al servidor local que también es utilizado para el desarrollo del diagrama Stack de la aplicación web del sistema de riego.

Para la primera prueba lo necesario es tener ya instalado Node.js, un editor para Js, en este caso se utilizó Atom y la librería express. Dicha librería tiene como función principal la creación de aplicaciones web.



```
app.js
1 var express = require("express"); //solicitud de la librería para la creación de la app
2 var app = express(); //variable app guardará la variable express
3 app.get("/", inicio)//URL que muestra el mensaje de la función inicio
4
5 function inicio(peticion, mensaje)//función donde la web hace petición al servidor, y este manda mensaje
6 {
7     mensaje.send("Aplicacion web");//mensaje del servidor
8 }
9
10 app.listen(8989); //puerto del servidor utilizado para encontrar la página web que mostrará el mensaje
11
```

Figura.2. Código en JavaScript para la prueba del servidor.

La ejecución del código fue mandada a llamar por medio de la consola, para esto es importante ingresar la dirección correcta de la carpeta del archivo.js, y tenerlo junto a la carpeta node_modules, la cual fue creada tras la instalación de la librería ‘express’.

```
[MacBook-Air-de-Agustin:~ jay-romo$ cd Documents
[MacBook-Air-de-Agustin:Documents jay-romo$ cd Servidor:cliente
[MacBook-Air-de-Agustin:Servidor:cliente jay-romo$ node app.js
```

Figura.3. Ejecución de código de la Figura.2 mediante la consola.

Después de la ejecución, se procede a ingresar el servidor local preestablecido junto con el puerto declarado en el archivo.js. “localhost:8989” es la manera sencilla para encontrar el mensaje ‘Aplicación web’ dentro del servidor web.

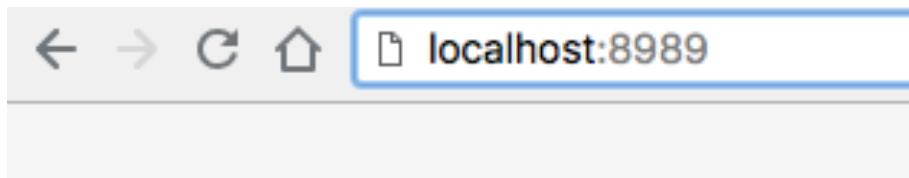


Figura.4. Servidor local.

Finalmente, al ver el mensaje dentro del servidor local de la Figura.4. Queda como prueba final de que el servidor web está corriendo correctamente.



Figura.5. Mensaje en Servidor web mediante Node.js JavaScript y express.

Como un dato curioso, al ingresar en un dispositivo móvil la dirección IP de la computadora sobre la que se está trabajando, y añadiendo el puerto, se mostrará el mismo mensaje, para que esto funcione, además de conectarse a la dirección IP de la computadora, el dispositivo móvil deberá estar conectado a la misma red wi-fi.

La fórmula final para lograr la conexión es: #direcciónIP:8989 desde cualquier navegador en el dispositivo con la misma red wi-fi.

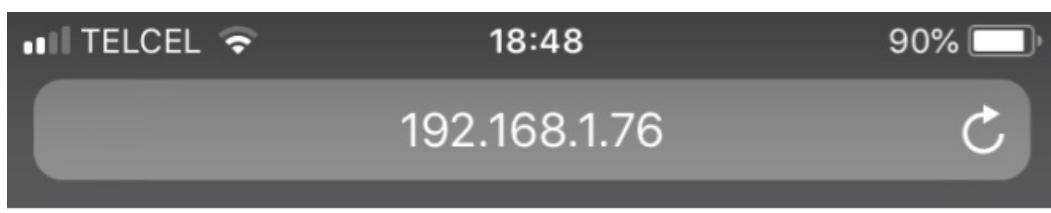


Figura.5. Mensaje en Servidor web desde dispositivo móvil.

2.2 Modelo Esquemático

Siguiendo entonces los patrones básicos para la creación de un servidor local, con la ayuda de 'express', se procede a crear el modelo esquemático de la aplicación para el control del hardware. Lo primero que se pretende, dentro del contexto del sistema inteligente, es la captura y lectura de datos importantes para llevar a cabo una correcta automatización.

La lectura de datos fue llevada a cabo mediante sensores. Los sensores previamente mencionados y definidos para la utilización en el diseño del proyecto, constaron de: un sensor basado en fotoceldas para la captura de la luz, un sensor DHT11 para medir temperatura y humedad en el ambiente y finalmente un sensor fc-28 para dar respuesta sobre si existe humedad de suelo o no. Los mencionados sensores serán mostrados gráficamente y se explicarán sus métodos de uso y ejecución en la parte de componentes dentro del capítulo 3.

Mediante el uso de los mencionados sensores, es como se llevó a cabo la prueba de captura e interpretación de datos dentro de la aplicación web.

Es en esta fase de desarrollo en donde el framework socket.io entra a jugar un papel fundamental, ya que este basa sus funciones en crear esa necesaria conexión entre el servidor y el cliente. Los ejemplos anteriores se basan únicamente en llevar a cabo la ejecución de un código de Js dentro de un servidor local, pero no existe una interfaz gráfica con la cual comenzar a interactuar como usuario, y es ese precisamente el problema que se resolverá a continuación, entonces para esto es que se necesita el framework socket.io y HTML para el desarrollo de la interfaz gráfica y la conexión de esta con el servidor respectivamente.

Si bien 'express' funciona para la creación de aplicaciones web, pero no es suficiente para llevar a cabo una conexión entre el servidor donde se llevó a cabo la aplicación con la interfaz para la interacción con el usuario, entonces teniendo esto claro, se realizó un nuevo código en Js en el que además de crear un mensaje que se muestre en el servidor, se mostraron los datos obtenidos por los sensores dentro de una página web con mejor forma que la observada en las figuras anteriores.

```
3 var http = require('http');
4 var express = require('express');
5 var socketIo = require('socket.io');
6 var app = express();
7 var server = http.createServer(app);
8 var io = socketIo.listen(server);
9
10 app.use(express.static(__dirname + '/public'));
11 server.listen(3000, () => console.log('Initializing server on port 3000'));
12
```

Figura.6. Herramientas que darán inicio para la creación de la aplicación en el puerto 3000.

La Figura.6. Muestra el inicio del nuevo código donde se solicitarán las librerías necesarias y previamente instaladas desde la terminal para crear un servidor en el puerto 3000 del ejemplo, cabe mencionar que el número de puerto en realidad no es importante si en un ejemplo anterior se utilizó un puerto distinto, es para demostrar que no importa en absoluto el número de puerto.

2.3 Lectura de datos

El origen dentro de la realización de la lectura de datos, estará siendo llevado a cabo en el IDE de Arduino, ya que se está utilizando esta tecnología con la que los mencionados sensores crean una gran interacción. Por lo tanto fue el IDE de Arduino el que proporcionó a Js los datos que obtuvo de cada uno de los sensores, para esto, lo primero fue ejecutar el código desde las herramientas que Arduino brindó.

```
/*SENALES*/
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11

int led = 13;
int foto;
int ilum;
const long A= 1000;
const int B= 15;
const int Rc= 10;
int fc = 4;
DHT dht(DHTPIN, DHTTYPE);
void setup()
{
    pinMode(fc,INPUT);
    pinMode(led,OUTPUT);
    Serial.begin(9600);
    dht.begin();
}
void loop()
{
    digitalWrite(led,LOW);
    delay(5000);
    int h = dht.readHumidity();
    int t = dht.readTemperature();
    if (isnan(h) || isnan(t) ) {
        Serial.println("Error obteniendo los datos del sensor DHT11");
    }
}
```

```

int hic = dht.computeHeatIndex(t, h, false);
/////////luz///////////
foto = analogRead(A0);
//ilum = ((long)foto*A*10)/((long)B*Rc*(1024-foto));
ilum = analogRead(foto);
//Serial.print("Luz: ");
//Serial.print(ilum);

/////////luz dht11 y fc-28///////////
if(ilum > 256 && ilum < 512)
{
  if(digitalRead(fc) == HIGH)
  {
    Serial.print("Humedad: ");
    Serial.print(h);
    Serial.print(" %");
    Serial.print(" Temperatura: ");
    Serial.print(t);
    Serial.print(" *C ");
    Serial.print(" Luz: Poca.");
    Serial.println(" Humedad de suelo: si");
  }
}

```

Figuras.7 y 8. Obtención de datos desde el IDE de Arduino.

La Figura.7 y 8. Muestran el código para la lectura y obtención de datos de cada uno de los sensores implementados en el sistema. Toda esta información fue transmitida al código de Js haciendo lectura del puerto donde la tarjeta electrónica de Arduino la estuvo enviando.

```

14 var Serialport = require('serialport');
15 var ReadLine = Serialport.parsers.ReadLine;
16
17 var port = new Serialport('/dev/cu.usbmodemFA131', {
18   baudRate: 9600
19 });

```

Figura.9. JavaScript toma los datos de Arduino desde el puerto /dev/cu.usbmodemFA131.

La Figura.9. Muestra cómo se logra la obtención de datos de Arduino que fueron mostrados en la aplicación, mediante el mismo rango de baudios para la comunicación serial se logra la muestra de los en la terminal haciendo llamar al código con node.js, los datos son mostrados tal cual como lo es en caso del IDE al utilizar el monitor serial. Para la obtención de la dirección del puerto, como es mostrado en la línea 17 del código, es importante tener en cuenta que sistema operativo se está usando, ya que dependiendo de eso la dirección cambia, se debe proceder desde el IDE de

Arduino a Herramientas, posteriormente Puerto y finalmente copiar la dirección del puerto USB tal cual se muestra. A continuación se presentan las fases finales para la obtención de datos del Arduino y su aspecto en la interfaz gráfica para la visualización del usuario.

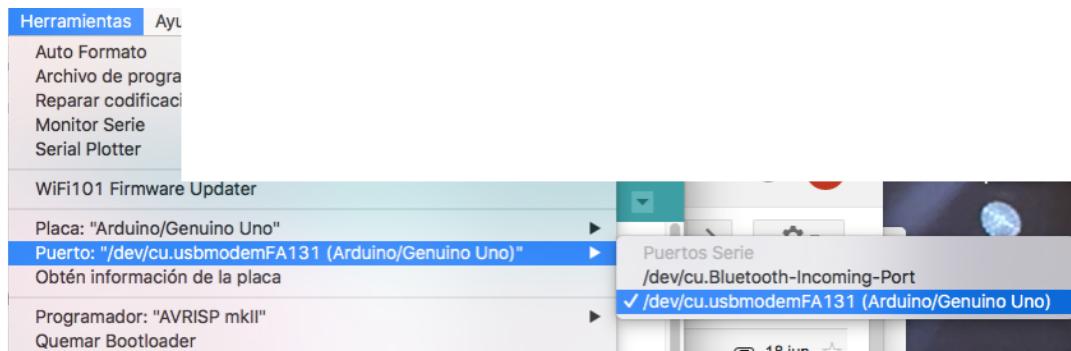


Figura.10. Manera de obtener la dirección exacta del puerto de trabajo en Arduino.

```
25 });
26 parser.on('data', function(data){
27   console.log(data);
28   io.emit('temp_hum', data);
29 });
30 port.on('err', function(err){
31   console.log(err);
32 })
```

Figura.11. Solicitud de impresión de datos en la terminal.

```
[MacBook-Air-de-Agustín:Server jay-romo$ node index.js
Initializing server on port 3000
Humedad: 45 % Temperatura: 24 *C Luz: Máxima. Humedad de suelo: no
Humedad: 45 % Temperatura: 24 *C Luz: Máxima. Humedad de suelo: no
Humedad: 45 % Temperatura: 24 *C Luz: Máxima. Humedad de suelo: no
```

Figura.12. Impresión de datos obtenidos por los sensores dentro de la terminal.

Al reproducir el código desde la terminal, mediante el comando `node código.js`, se muestran en la terminal tal cual como en el monitor serial de Arduino los datos obtenidos por los sensores, ahora sólo falta llevar a cabo el diseño de la página donde el usuario podrá visualizar esta información.

Es en esta fase donde se usará HTML, el lenguaje no de programación creado para el diseño gráfico de las aplicaciones web.

```
1  <!DOCTYPE html>
2  <html lang="en" dir="ltr">
3      <head>
4          <meta charset="utf-8">
5          <title>App Web</title>
6          <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
7          <style>
8
9      .btn {
10         width: 200px;
11         padding: 15px 32px;
12         align-items: center;
13     }
14     body
15     {
16         color: #F7F9F9;
17         font-family: comic;
18     }
19     </style>
20 </head>
21 <body background="circuito2.jpg">
22     <div class="container theme-showcase" role="main">
23         <h2>Sistema de riego</h1>
24         <div class="row">
25             <div class="col-md-1 col-md-offset-1">
26                 <p> <card id="temp_humi"></card>
27             </div>
28         </div>
```

Figura.13. Código HTML para diseño de la interfaz gráfica.

En los códigos de HTML, se llevan a cabo cuáles fueron los elementos vitales para el entendimiento de lo que se observa en pantalla al ingresar a su página como el título, algunos encabezados y textos adicionales, el color y tamaño de la letra, color o imagen de fondo, mediante `<etiquetas>`, la forma en como HTML realiza sus múltiples acciones para dar un buen y agradable diseño a las páginas.

Es también en HTML donde se crean los “id” atributos globales que definen un identificador único que no debe repetirse en todo el documento. Cumple como objetivo identificar el elemento al vincularlo en Js. Su valor se basa en una cadena de caracteres. El id de este ejemplo contiene los datos capturados del puerto USB los cuales por medio de un nuevo código en Js fueron mostrados dentro de la interfaz.

```
21      <div class="row">
22          <div class="col-md-1 col-md-offset-1">
23              <p> <card id="temp_humi"></card>
24          </div>
25      </div>
26      <script src="/socket.io/socket.io.js" charset="utf-8"></script>
27      <script src="/main.js" charset="utf-8"></script>
28  </body>
29 </html>
```

Figura.14. Declaración del atributo id “temp_humi” que contiene los datos.

La razón por la que se está utilizando la etiqueta card es por la forma en cómo se desea que los datos sean mostrados dentro de la interfaz, el documento HTML solicita directamente con el archivo main.js que es el que asigna el valor de todos los datos (previamente guardados en una variable data en el anterior código) al identificador temp_humi.

```
1 var socket = io();
2
3 var temperatureDisplay = document.getElementById('temp_humi');
4
5 socket.on('temp_hum', function (data) {
6     console.log(data);
7     temp_humi.innerHTML = `${data}`;
8 });
9
```

Figura.15. Código que asigna el valor data al atributo id y lo manda a imprimir en la terminal.

El pequeño código main.js del identificador debe adherirse a la conexión con la interfaz y el código anterior, es por eso que se asigna a la variable io, la cual previamente fue declarada para guardar esa comunicación con el servidor por medio de socket.io como es mostrado en la Figura.6.

A continuación se presenta la figura que revela como queda el diseño de la aplicación web mostrando los datos de los sensores tomados desde el puerto USB por Javascript y representados en su interfaz gráfica con la utilización de HTML.

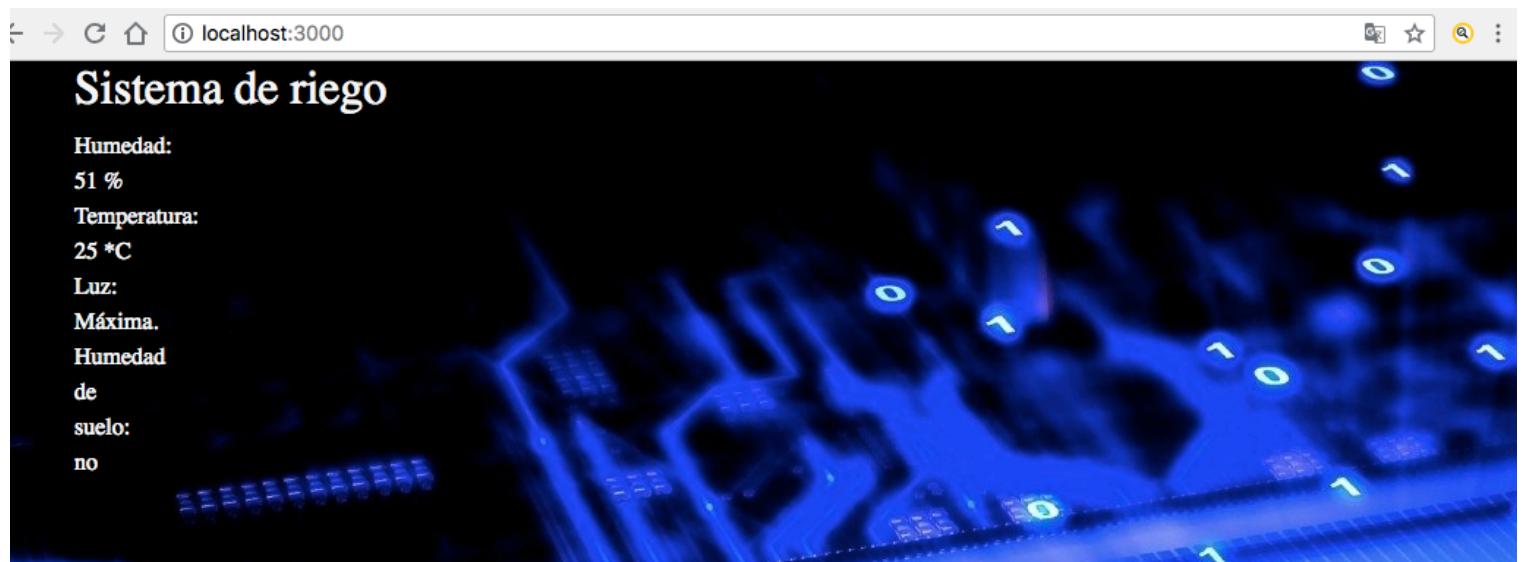


Figura.16. Interfaz gráfica para la muestra de datos con la dirección localhost:3000.

Capítulo 3.

Realización, Implementación y Pruebas

A continuación se muestran de forma detallada los componentes del sistema, así como sus funciones específicas, la esquematización y una visión final en el diseño de la aplicación web donde se estará llevando del registro de los movimientos realizados por la automatización del prototipo.

3.1 Sistema, etapas y componentes

Como es habitual, un sistema fundamental debe estar conformado por tres etapas básicas, una entrada, una etapa de procesamiento y una salida. Definase sistema en este concepto como un conjunto de elementos que realizan un proceso manipulando la señal adquirida.

La etapa de entrada se encargará de la adquisición de múltiples señales analógicas y digitales, es por ello que la etapa inicial consta habitualmente de sensores que son los encargados de captar esas señales. En el modelo del prototipo, se han implementado tres sensores para su etapa de entrada, el primero es un sensor analógico que capta la luz del entorno mediante una fotocelda, el segundo es un sensor de clase digital que capta niveles de humedad del ambiente en porcentaje y temperatura ambiente en grados centígrados. Y el último es un sensor híbrido(analógico y digital)que capta la humedad en la tierra.

Estos tres sensores midieron las señales necesarias para obtener los parámetros específicos que establecieron si eran necesario o que tan necesario era regar las plantas del jardín. Luz, humedad y temperatura ambiente y humedad del suelo, los parámetros sobre los que se llevó a juicio el cuidado y mantenimiento de las plantas, todo esto llevado a una lógica de programación para hacer que la automatización del sistema de riego contemplara bajo qué condiciones climatológicas y de suelo se debió actuar.

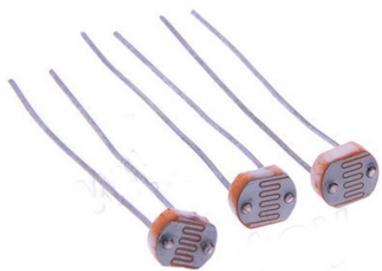


Figura.17. Fotoceldas.

La photocelda o fotoresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz. Por medio de este criterio, se parametrizará la medición de luz en máxima, media y poca.

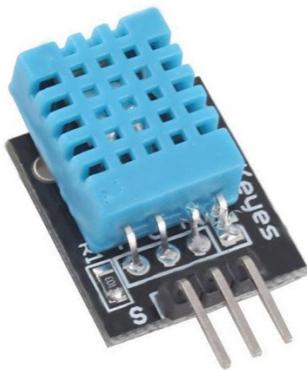


Figura.18. Sensor DHT11.

El DHT11 es un sensor para medir humedad y temperatura ambiente, de clase digital, el campo más común para su uso es debido a sus características de trabajo la estación metereológica, basándose en su propia librería es como puede aprovecharse al máximo su modo de empleo para obtener los datos esperados.

Consta de una alimentación de 3.5 v a 5 v, un consumo de 2.5mA, puede medir un rango de temperatura de 0°C a 50°C con una resolución de 8 bits por grado y precisión de 25°C ± 2°C. En humedad tiene un rango de 20% RH a 90%, precisión entre 0°C y 50°C ± 5% RH, con resolución de 1% RH.

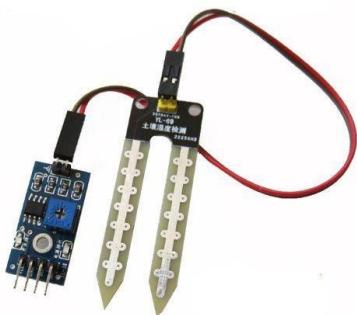


Figura.19. Sensor FC-28.

Mencionado en algunas páginas de soporte como Higrómetro de suelo, un sensor que mide la humedad en el suelo, es una de las herramientas más aplicadas para automatización de sistemas de riego que detectan cuando es necesario activar algún tipo de riego.

Mediante la variación de su conductividad puede medir la humedad aunque no posee una precisión lo suficientemente exacta para realizar una medición completa.

El FC-28 se distribuye con una placa de medición estándar que permite obtener el resultado como valor analógico o como una salida digital, activada cuando la humedad ha superado un rango significativo de humedad.

El valor más concreto dependerá de tipo de suelo y la presencia de elementos químicos, como fertilizantes. Al implementar esta herramienta es importante que el usuario conozca el nivel de humedad que cada una de sus plantas necesita.

Después de conocer de una forma física y conceptual los sensores que formarán parte de la entrada de datos, la siguiente etapa dentro del sistema es la captura e interpretación de los mismos. Un procesamiento de datos se lleva mediante la etapa de control, la cual es llevada a cabo con una tarjeta electrónica Arduino UNO.

La adquisición de datos es la que permitirá a los actuadores trabajar mediante un lenguaje de programación que se encargará de manipular toda información digital y

analógica que reciba con sentencias lógicas para dar conclusión al sistema por medio de los componentes encargados de efectuar la tarea de riego.



Figura.20. Tarjeta electrónica Arduino UNO.

La tecnología de Arduino provee muchas facilidades para proyectos de esta magnitud, ya que su manera de programar es sencilla y cuenta con múltiples plataformas de soporte y ayuda en internet, así como ejemplos, y múltiples dispositivos que son enteramente compatibles con sus tarjetas.

La Figura.20. Muestra el Arduino UNO, utilizado para las fases tanto de diseño digital y analógico como la etapa de control, el “cerebro” del sistema tiene la capacidad de leer, interpretar y efectuar tareas de datos analógicos y digitales, además de que proporciona alimentación de voltaje mediante sus puertos de energía.

Tras concluir con la etapa de control, todas las señales obtenidas deben ser interpretadas por la última etapa del sistema básico, la salida. Por medio de dos actuadores la salida constará en proporcionar el riego adecuado para el cuidado de las plantas, todo esto acompañado de las investigaciones previas llevadas a cabo por el usuario sobre que tipo de cuidado debe tener cada una de sus plantas.

La etapa de salida esta compuesta por dos actuadores, un servo y una mini bomba de agua, el servo realiza la tarea de direccionar la manguera o aspersor para brindar un riego adecuado en toda el área, o zona de riego, la bomba realiza las tareas de riego continuo y de bombear no más de lo necesario para cada planta. Todo esto aunado a las previas condiciones establecidas por las medidas en la etapa de inicio.



Figura.21.Micro Servo 9g Tower Pro.

El servo o servomotor es un motor eléctrico que consta de dos características especiales. La primera permite mantener una posición indicada, siempre que se encuentre dentro del rango de operación del dispositivo. La segunda permite controlar la velocidad de giro, es posible manipularlo de tal manera que antes de cambiar de posición, espere un tiempo.

El tipo de servo mostrado en la Figura.21. Tiene algunos aspectos a considerar, su ángulo permite hacer un barrido entre -90° y 90°. Dando un ángulo de 180°.

Independientemente de que el servo sea capaz de moverse con resolución superior a 1 grado, ese será su máximo grado de resolución posible de conseguir debido a las limitaciones de la señal PWM^{cita} que puede generar la tarjeta electrónica de la Figura.20.

La señal PWM le da la función al dispositivo con un pulso de trabajo entre 1 ms y 2 ms y con un periodo de 20 ms (50 Hz). Todo esto hace referencia a la velocidad máxima con la cual se puede mover el motor con la tarjeta electrónica.



Figura.22. Mini bomba de agua.

Las bombas de agua son máquinas hidráulicas que permiten incrementar la energía cinética de un caudal de agua. Con ellas se puede controlar todo un equipo de bombeo por medio de procesadores, en su mayoría automáticos.

Para todo establecimiento con sistemas de abastecimiento y depuración de agua y sistemas de climatización, las bombas de agua son de fundamental implementación. Incluso en el ámbito de la industria, forman parte de múltiples áreas que impulsan todo tipo de fluidos.

Pero ¿qué es lo que realmente hace que una bomba de agua funcione?. La bomba de agua está compuesta por un motor eléctrico acoplado a un rodete, formado por álabes. Los álabes son paletas curvas adheridas al rodete que cuando giran, proporcionan energía al fluido. La forma de los álabes determina la cantidad de energía transmitida al fluido y el grado en como esa energía se reparte entre velocidad o presión.



Figura.23. Vistazo al interior de una mini bomba de agua.

La mini bomba de la Figura.22. Es de tipo sumergible, con una alimentación de 3-5 v, corriente eléctrica de 100-200 mA y un caudal: 70-120 L/H.

3.2 Sistema de riego por aspersión casero

Este tipo de sistema consta en que la aplicación de riego cubra toda un área, o bien, todo el contorno de una maceta, es un sistema plenamente utilizado para riego de plantas de jardín y para riego de campos agrícolas. Es posible de ser utilizado en terrenos llanos como en zonas con elevaciones o depresiones de terreno.



Figura.24. Sistema de riego por aspersión tradicional.

Cuando el riego de plantas se aplica manualmente, no siempre es sencillo tener en cuenta el criterio para el cuidado del agua o la incertidumbre de saber si la planta necesita más o menos agua de la que se está aplicando.



Figura.25. Sistema de riego manual tradicional.

Ahora, no todos los hogares son candidatos a implementar un sistema de riego por aspersión casero en sus jardines debido a la escasa cantidad de pasto, plantas o espacio, así como tampoco contar con las condiciones adecuadas para la inserción de tuberías y aspersores, por lo que el diseño físico del sistema pretende ser más amigable para estos casos.

Este nuevo sistema de riego casero pretende además de medir parámetros metereológicos y de suelo, tener en cuenta el cuidado del agua y el buen mantenimiento de las plantas, pero como ya se ha mencionado anteriormente, esto último es tarea del usuario.

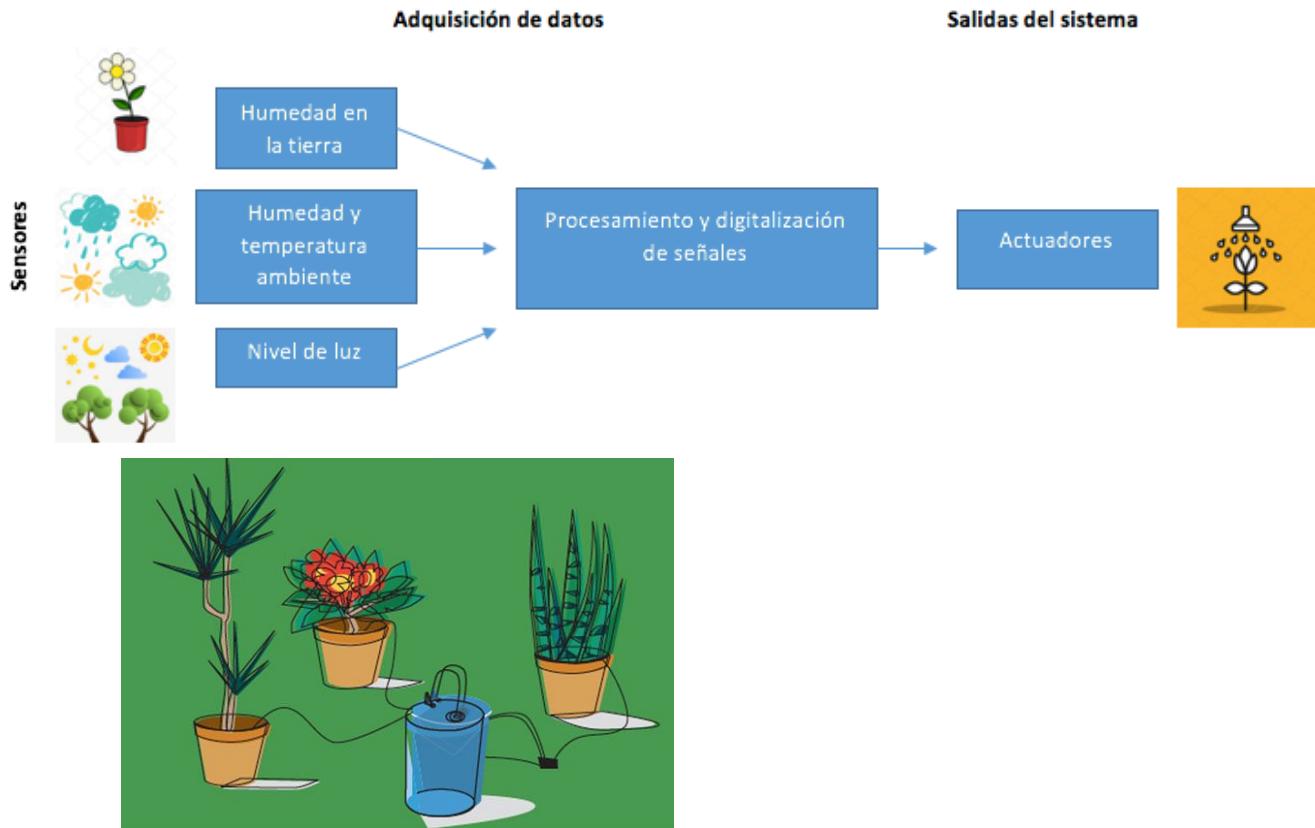


Figura.26. Contemplado diseño animado para sistema de riego automático.

Figura.27. Diagrama en bloques del Sistema de Riego Casero Automático.

3.3 Emisión y Recepción

Ya que el sistema está compuesto por tres sensores de diferentes características y dos actuadores que requieren un entorno propio para su correcta ejecución, se llevó a cabo la tarea extra de dividir el mencionado sistema en un emisor y un receptor, con la finalidad de que ambas partes pudieran trabajar en conjunto desde bases separadas y brindar un correcto uso.

La comunicación bluetooth es un medio bastante útil, eficaz y fácil de emplear para estos casos. Gracias a la comunicación bluetooth se puede tener un entorno de automatización adecuado sin riesgo de fallas, siempre y cuando se lleven las configuraciones adecuadas para los dispositivos maestro y esclavo y las conexiones debidas.

El modo emisor contó con su dispositivo modular bluetooth, los tres sensores y el actuador servo. Es importante entender que este actuador realizó la tarea de dispersar la manguera de agua por el contorno frontal de una maceta para evitar que el chorro de agua fuera arrojado en un solo punto, pero aunque está fue la única tarea del servo, es muy importante estar en perfecta sincronía con la bomba de agua.

Es importante también hacer mención que los datos de los sensores no fueron enviados por bluetooth, la manera de obtención de datos ya se mencionó en el capítulo anterior. El módulo de bluetooth maestro le dice al esclavo cuando debe activar la bomba de agua, al mismo tiempo que se debe activar el servo, así, cuando el agua fluya a través de la manguera, es distribuida por los tres extremos de riego, izquierda, centro y derecha.

Así entonces, el modo receptor está compuesto únicamente del módulo esclavo y la bomba de agua conectada a la manguera, y esta última direccionada por el servo.

Este complemento adicional fue implementado debido a las fallas que presentaba el servo durante su ejecución activándose al mismo tiempo que la bomba de agua. En plena fase de pruebas de ambos actuadores, el usuario se percató de una inusual actividad por parte del servo, si bien, tanto el servo como la bomba deben trabajar al unísono, el servo no tenía la capacidad de funcionar correctamente cuando la bomba era activada, es por esto que se tomó la decisión de hacerlos funcionar por separado pero conectados dentro del mismo sistema de forma inalámbrica como solución.

3.3.1 Configuración maestro y esclavo

El dispositivo utilizado en la etapa de comunicación para la solución de función de los actuadores fue el módulo de bluetooth HC-05, un modelo que puede configurarse como emisor y receptor de datos. La tarjeta electrónica Arduino UNO es de gran utilidad para llevar a cabo las configuraciones.



Figura.28. Módulo de bluetooth HC-05.

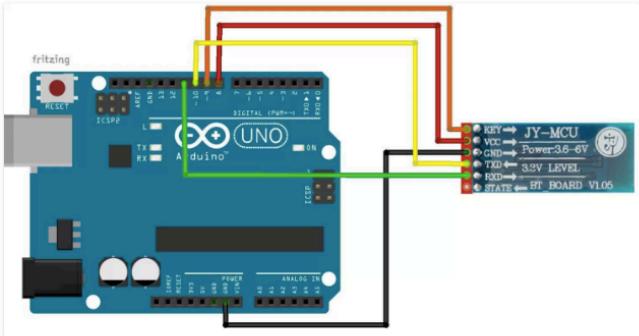


Figura.29. Manera de configurar el módulo con Arduino.

https://naylampmechatronics.com/blog/24_configuracion-del-modulo-bluetooth-hc-05-usa.html

El módulo debe entrar en la modalidad de comandos AT^{cita}. Estos comandos brindan un cambio de nombre al establecido de fábrica, cambio de contraseña, muestran el modo en que el dispositivo se encuentra, sea esclavo = 0 ó maestro = 1, y revelan la dirección brindada por el esclavo para la comunicación entre el emisor y el receptor.

Por lo regular estos modelos cuentan con un pulsador del lado opuesto del pin Enable o Key. El pulsador debe ser presionado y no soltarlo hasta después de realizar la alimentación como se muestra en la Figura.29. El pequeño led de la esquina inferior izquierda comienza a parpadear lentamente como indicación de que ya se encuentra en la modalidad de aceptar comandos AT.

Posterior a lo anteriormente mencionado, se debe ejecutar un pequeño código en el IDE de Arduino para la realización de la configuración. Después de la ejecución del código, se ingresan los comandos AT requeridos en el monitor serial para dar por terminada la configuración del módulo en ambos, maestro y esclavo.

```

#include <SoftwareSerial.h>
SoftwareSerial BT1(10, 11); // RX | TX
void setup()
{
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    digitalWrite(9, HIGH);
    delay (500) ;
    Serial.begin(9600);
    Serial.println("Configuración del módulo HC-05");
    digitalWrite (8, HIGH);
    Serial.println("Lectura de comandos AT:");
    BT1.begin(38400);
}

void loop()
{
    if (BT1.available())
    {
        Serial.write(BT1.read());
    }
    if (Serial.available())
    {
        BT1.write(Serial.read());
    }
}

```

Figura.30. Código de configuración del módulo HC-05.

La segunda línea del código dice que los puertos 10 y 11 de la tarjeta Arduino representan recepción y transmisión (RX, TX) respectivamente. Estos puertos deben intercalarse con los de recepción y transmisión del módulo, ya que de lo contrario no funcionará la configuración.

	COMMAND	FUNCTION
1	AT	Test UART Connection
2	AT+RESET	Reset Device
3	AT+VERSION	Query firmware version
4	AT+ORGL	Restore settings to Factory Defaults
5	AT+ADDR	Query Device Bluetooth Address
6	AT+NAME	Query/Set Device Name
7	AT+RNAME	Query Remote Bluetooth Device's Name
8	AT+ROLE	Query/Set Device Role
9	AT+CLASS	Query/Set Class of Device CoD
10	AT+IAC	Query/Set Inquire Access Code
11	AT+INQM	Query/Set Inquire Access Mode
12	AT+PSWD	Query/Set Pairing Passkey
13	AT+UART	Query/Set UART parameter
14	AT+CMODE	Query/Set Connection Mode
15	AT+BIND	Query/Set Binding Bluetooth Address

Figura.31. Comandos AT básicos para configuración maestro/esclavo.

3.4 Implementación final y pruebas

La última fase dentro del desarrollo y diseño del sistema fue la implementación final del dispositivo realizando las labores solicitadas, así mismo, el dispositivo fue sometido a diversas pruebas para riego de al menos tres tipos de plantas distintos en diferentes bases o macetas.

Del mismo modo, la aplicación web fue sometida a la prueba final de mostrar en pantalla los parámetros ambientales entregados por los sensores que componen el subsistema emisor y la lectura de si el sistema se encontraba trabajando o en reposo.

La implementación del sistema para la realización de pruebas fue llevada a cabo en una casa particular ya que como recordatorio el primer prototipo de este sistema está pensado en funcionar dentro de un hogar como una de las múltiples tareas que un dispositivo inteligente puede realizar en una casa.

Tras la implementación final, se han notado los detalles a modificar en el diseño del ensamblaje del sistema, debido a que se desea que el sistema sea lo más portátil posible, es importante hacer algunas ligeras modificaciones en cableado, métodos de alimentación del sistema y algunos elementos que fueron utilizados para el ensamblaje mismo.



Figuras.31 y 32. Prototipo subsistema emisor y subsistema esclavo con disparador de agua.

A continuación se muestran las figuras tomadas en la fase de pruebas.

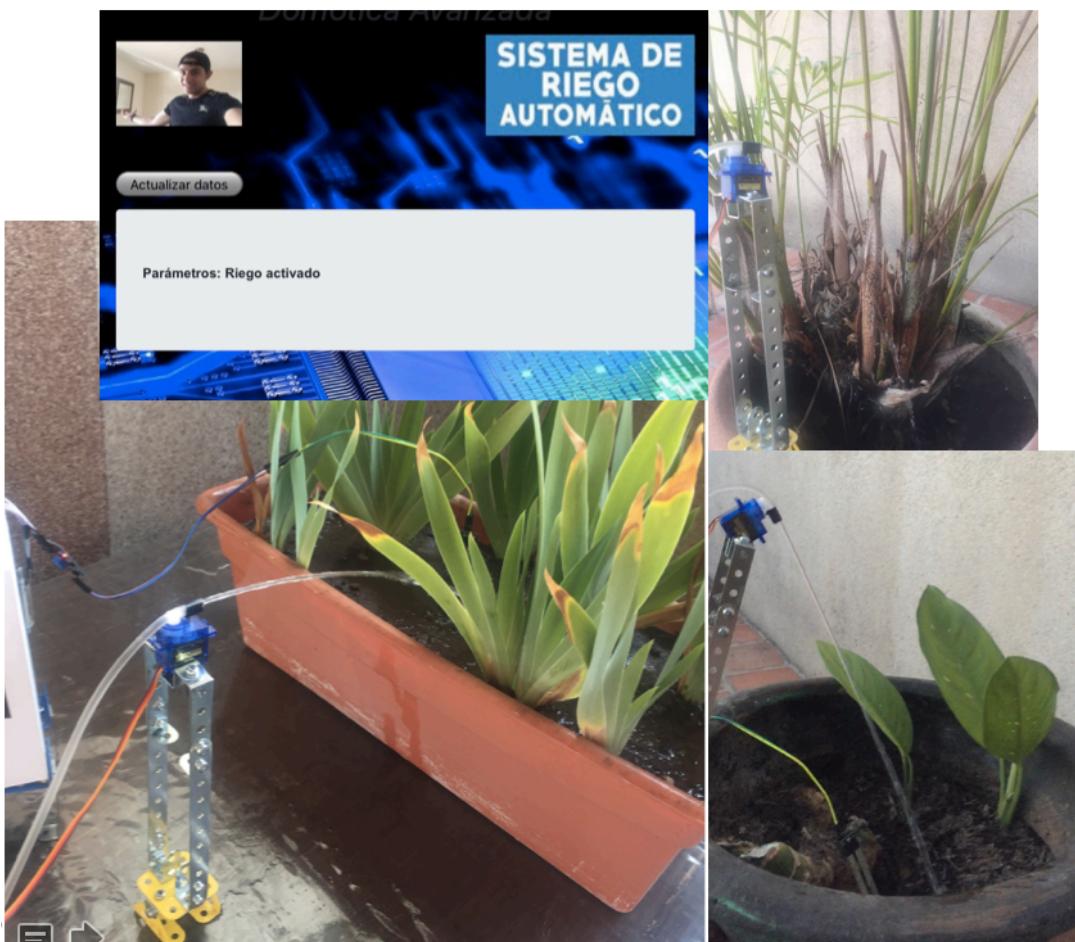


Figura.33. Sistema activado.

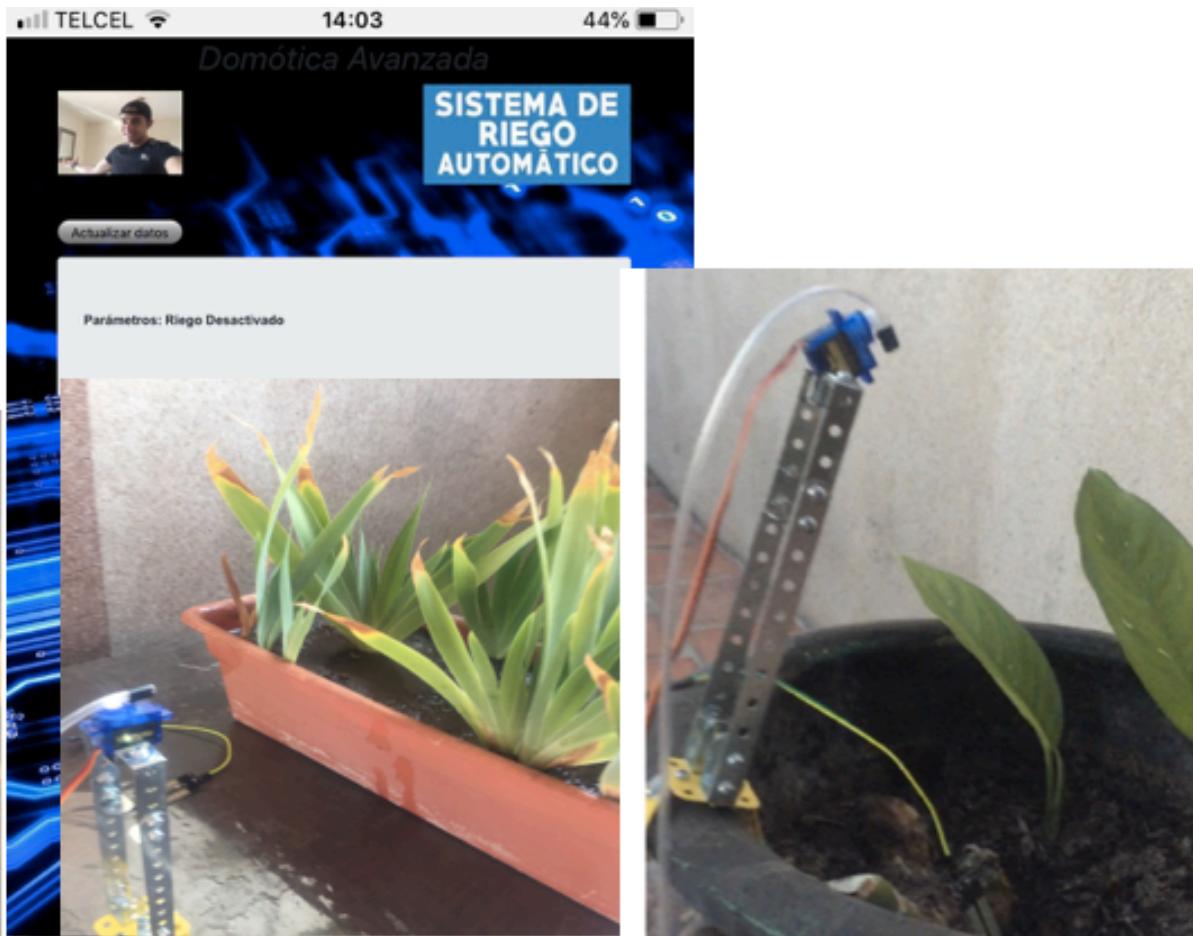


Figura.34. Sistema desactivado.

Conclusión.

El desarrollo de este prototipo ha otorgado un gran conocimiento en múltiples herramientas para el enfoque en el desarrollo de software. Desde un nuevo lenguaje de programación hasta diseñar páginas web y correrlas dentro del servidor, y todo esto hace que al saber como crear aplicaciones web en tiempo real abre las puertas para la creación de nuevos proyectos de automatización y el perfeccionamiento de los mismos.

La fase de pruebas detalló que los objetivos específicos fueron obtenidos, el sistema dio un gran funcionamiento en el cuidado diario de plantas de jardín y todos los registros y movimientos fueron monitoreados sin interrupción por la aplicación web, todo esto en conjunto da un formidable inicio para la creación de futuros dispositivos inteligentes en la gama de internet de las cosas para el mejoramiento de la calidad de vida en el hogar, sin mencionar que existen otras áreas en las que pueden ser implementados múltiples tipos de dispositivos automatizados como hospitales y la misma industria.

El conocimiento de cómo implementar tecnologías para el desarrollo de las casas inteligentes brinda un amplio contexto de cómo crear e implementar todos los sistemas posibles para la obtención de los resultados deseados. La domótica tal vez no es un campo de mucha exploración o desarrollo en ciertos países, pero lo que se desea con este tipo de prototipos es dar un motivador inicio en el desarrollo de este campo.

El siguiente paso para todo esto es dar un mejoramiento a los prototipos ya existentes, y no solo en cuestión a lo físico o al hardware sino también contemplar las normas de calidad aplicables para que el sistema sea digno de ser llevado a su producción para posterior comercialización.

El mundo digitalizado de hoy en día obliga a modificar el estilo de vida, tal vez no en el hecho de hacerse dependientes de la automatización pero si en el hecho de contar con ella como una mano extra para ciertas tareas y labores, debido a esto, es importante comenzar a implementar oportunidades de desarrollo y diseño de estas tecnologías que son una útil inversión, que fomentarán el desarrollo tecnológico de las naciones y un incremento en índices de desarrollo y calidad de vida.

Recomendaciones.

Para el futuro diseño de dispositivos automatizados, la recomendación principal es obtener un conocimiento absoluto de las herramientas para el diseño de software que se desean implementar estando conscientes de sus alcances y limitaciones, ya que dependiendo del tipo de dispositivos que se desea crear y el área donde se desea poner en práctica, estas tecnologías pueden revelar resultados alternos, incluso tal vez resultados no contemplados.

La capacitación para el conocimiento y dominio de las nuevas tecnologías para internet de las cosas, es un paso que no debe ser omitido, ya que se habla de un campo bastante extenso del que siempre se aprenden cosas brillantes.

La recomendación respecto al proyecto está basada en como implementar las mejoras, se desea que el sistema sea lo mas portátil posible y que no dependa de una computadora externa para la parte del monitoreo de datos, un intercambio de tecnología wifi por la de bluetooth puede mejorar la comunicación serial y la modificación en el aspecto del ensamblaje de los subsistemas debe ser un grado más profesional ya que se tiene como objetivo a medio plazo llevar la comercialización de este como prototipo inicial.

El stack conformado por Node.js Javascript y HTML proporcionó un avance considerable en el diseño de la aplicación web pero se limita a la tecnología alámbrica, la automatización de sistemas debe constar en modelos que puedan ser ejecutados sin necesidad de dispositivos externos a menos que sea de forma inalámbrica, la recomendación es explorar que tecnologías de servidor pueden hacer frente a esta problemática y que dispositivos cuya naturaleza sea la comunicación serial pueden integrarse al todo el modelo esquemático digital para lograr resolver el mencionado detalle.

Una vez más la capacitación y total conocimiento de las tecnologías para el desarrollo en internet de las cosas se hacen presentes como la principal herramienta para el buen desarrollo en calidad de estos dispositivos y sistemas inteligentes. La realización de pruebas en diversas condiciones a las que se puede someter el sistema es otra manera efectiva de obtener una buena calidad al ver la respuesta que dan, y así evaluar con certeza la necesidad de ajustes y cambios.