

DISEÑO DE SISTEMA DE RIEGO AUTOMÁTICO PARA GERMINADORES DENTRO
DE INVERNADEROS

VICTOR MANUEL GALEANO LEÓN

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE TECNOLOGÍAS
INGENIERÍA MECATRÓNICA

2023

DISEÑO DE SISTEMA DE RIEGO AUTOMÁTICO PARA GERMINADORES DENTRO
DE INVERNADEROS

VICTOR MANUEL GALEANO LEÓN

Trabajo de grado para optar por el título de
TECNÓLOGO EN MECATRÓNICA

Director

M.Sc. HERNÁN ALBERTO QUINTERO VALLEJO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE TECNOLOGÍAS

INGENIERÍA MECATRÓNICA

2023

Nota de aceptación

DIRECTOR

ESTUDIANTE

DEDICATORIA

A Dios y a mis padres por su apoyo, que me permitieron realizar este trabajo.

AGRADECIMIENTOS

Son muchas las personas a quienes debo agradecer, pero en especial a HERNÁN ALBERTO QUINTERO VALLEJO, M.sc en Instrumentación Física, docente del programa Ingeniería Mecatrónica, quien con sus conocimientos aportó al desarrollo de este trabajo.

RESUMEN

El presente documento describe el diseño, construcción e implementación de un sistema de riego automático para germinadores dentro de invernaderos, que actúa con un Control Numérico Computarizado (CNC). Con el cual se mejoran tiempos de producción, costos y procesos de mantenimiento.

Para su implementación fue necesario realizar inicialmente un análisis de los elementos que los componen, así como las etapas de diseño en software CAD de cada una de las piezas.

En los apartados de programación y diseño electrónico, se puede observar el proceso de codificación para el funcionamiento óptimo del mecanismo, así como del monitoreo constante para que el desplazamiento en el sistema, buscando la precisión correcta con cada una de las coordenadas asignadas para el riego en un punto en específico.

Finalmente, se lleva a cabo el ensamble de la estructura mecánica, pruebas de funcionamiento y discusión con respecto a los resultados obtenidos durante el desarrollo del prototipo de riego automático.

JUSTIFICACIÓN

Resulta difícil cultivar semillas en germinadores, cuando están en ambientes que se encuentran a la intemperie o que no están controlados, lo que lleva a pérdidas de material (semillas) y de tiempo esperando su germinación dando, además, incertidumbre en el material germinado que tendremos para la siembra.

Los métodos convencionales para la germinación de semillas retrasan los procesos, además de ser inexactos en cuanto al riego que requiere cada tipo de semilla, dado que se realiza manualmente, esto hace que la cantidad no sea la óptima; Adicionalmente los tiempos de irrigación se ven afectados por diferentes tareas que debe realizar la persona encargada del riego, lo que hace improbable cuantas semillas germinadas se pueden obtener para su posterior siembra en el terreno o venta de semilla germinada.

La germinación dentro un invernadero permite resolver algunos problemas ya que se encuentra en un ambiente controlado, sin embargo, sigue teniendo el problema de la exactitud al momento de regar cada una de las semillas, como exceso de agua o poca cantidad de esta, obteniendo resultados que son un problema para el cultivo germinado porque podría pudrirse la semilla.

La problemática expuesta anteriormente podría ser mejorada o solucionada, con un sistema automático de riego, que dosifique la cantidad de agua necesaria de acuerdo con el tipo de semilla y, el tiempo de ella dentro del germinador, lo cual lleva a la siguiente pregunta de investigación:

¿Es posible diseñar y construir una máquina que permita dosificar, de manera automática, la cantidad de agua que se aplica a cada una de las semillas que se encuentran dentro de un germinador, con horarios y tiempos preestablecidos de acuerdo con el tiempo de germinación?

TABLA DE CONTENIDO

INTRODUCCIÓN	1
OBJETIVOS.....	2
OBJETIVO GENERAL.....	2
OBJETIVOS ESPECÍFICOS	2
1 MARCO CONCEPTUAL.....	3
1.1 INVERNADERO	3
1.2 AUTOMATIZACIÓN.....	3
1.3 GERMINADOR O SEMILLERO	4
1.4 SISTEMA DE RIEGO.	5
2 MARCO DE REFERENCIA	8
2.1 ASPERSIÓN:.....	8
2.2 GOTEO:	9
2.2.1 AQUA4D:.....	9
2.2.2 RIEGO POR NEBULIZACIÓN:	10
2.2.3 TWO VALVE:.....	12
3 MARCO NORMATIVO	13
4 SISTEMA DE RIEGO DE GOTEO AÉREO	14
4.1 SISTEMAS DE RIEGO AÉREO.....	14
4.1.1 SISTEMA DE RIEGO AÉREO POR GOTEO.....	14
4.2 REQUERIMIENTOS.....	15
4.3 EQUIPOS.....	16
4.3.1 BOMBA PARA SISTEMAS DE GOTEO.....	16
4.3.2 MANGUERA DE 6MM	17
5 DISEÑO MECÁNICO	18
5.1 ¿QUÉ ES UNA CNC?.....	18
5.2 CORE [X, Y]	18
5.2.1 PRINCIPIO DE OPERACIÓN:	18
5.3 DISEÑO CAD DE LA CNC PARA EL MECANISMO DE RIEGO.....	21
6 DISEÑO ELECTRÓNICO	28
7	28
7.1 COMPONENTES ELECTRÓNICOS.....	28
7.1.1 MÓDULO ESP32:.....	28
7.1.2 MOTOR PASO A PASO 17HS3401 Y DRIVER A4988:.....	28

7.1.3	TRANSISTOR FDD8580 SMD D-PACK:	30
7.1.4	REGULADOR 5V LDO LD1117 SMD:	31
7.1.5	FINALES DE CARRERA:	31
7.1.6	FUENTE 12V 5A:.....	33
7.2	ESQUEMA DE CONEXIONES	33
7.2.1	DISEÑO PCB	34
	PARA EL DISEÑO DE LA PLACA PCB, TENIENDO EN CUENTA LAS CONEXIONES ANTERIORES, SE UTILIZÓ EL SOFTWARE EAGLE CAD.....	34
	35
8	DESARROLLO DEL SOFTWARE PARA EL CONTROL DEL SISTEMA DE RIEGO	36
8.1	INTERFAZ DE USUARIO.....	36
8.1.1	ARDUINO IOT CLOUD.....	36
8.2	CODIFICACIÓN	40
	RESULTADOS.....	46
	DISCUSIÓN.....	51
	HERRAMIENTAS Y COSTOS	52
	CONCLUSIONES	54
	RECOMENDACIONES.....	56
9	REFERENCIAS.....	57
	ANEXOS.....	59

LISTADO DE IMÁGENES

<i>Ilustración 1 Ejemplo de invernadero.</i>	3
<i>Ilustración 2 Ejemplo gráfico de germinador o semillero.</i>	5
<i>Ilustración 3 Sistemas de riego.</i>	7
<i>Ilustración 4 Diferencias en cultivo con la implementación de Aqua4D.</i>	10
<i>Ilustración 5 Riego por nebulización.</i>	11
<i>Ilustración 6 Sistemas Two Valve.</i>	12
<i>Ilustración 7 Componentes principales de un sistema de riego por goteo.</i>	14
<i>Ilustración 8 Bomba para el sistema de riego aéreo por goteo.</i>	16
<i>Ilustración 9 Manguera neumática de 6mm.</i>	17
<i>Ilustración 10 Acople rápido PC6 01.</i>	17
<i>Ilustración 11 Disposición de las correas.</i>	19
<i>Ilustración 12 Accionamiento de las correas.</i>	19
<i>Ilustración 13 Mecanismo Core XY de referencia.</i>	20
<i>Ilustración 14 Estructura de CNC para sistema de riego por goteo aéreo.</i>	22
<i>Ilustración 15 Plano explosionado en Software CAD.</i>	23
<i>Ilustración 16 Base de la CNC.</i>	24
<i>Ilustración 17 Cubierta de motores.</i>	24
<i>Ilustración 18 Deslizador lateral.</i>	25
<i>Ilustración 19 Cubierta de poleas.</i>	25
<i>Ilustración 20 Polea impresa GT2.</i>	26
<i>Ilustración 21 Deslizador central.</i>	26
<i>Ilustración 22 Varilla de deslizador.</i>	27
<i>Ilustración 23 Módulo Wifi ESP32.</i>	28
<i>Ilustración 24 Motor Nema 17HS3401.</i>	28
<i>Ilustración 25 Driver A4988, para motor PAP.</i>	29
<i>Ilustración 26 Transistor FDD8580 SMD D-pack.</i>	30
<i>Ilustración 27 Regulador LD1117.</i>	31
<i>Ilustración 28 Final de carrera para CNC.</i>	31
<i>Ilustración 29 Adaptador de 12v 5A.</i>	33
<i>Ilustración 30 Esquema de conexiones circuito eléctrico de Sistema de Riego.</i>	34
<i>Ilustración 31 Trazo del esquema de conexiones en Eagle.</i>	35
<i>Ilustración 32 Layout de la PCB.</i>	35
<i>Ilustración 33 Ejemplo de la interfaz de usuario organizada en el Dashboard de Arduino IoT Cloud.</i>	36
<i>Ilustración 34 Suscripción Arduino Cloud.</i>	37
<i>Ilustración 35 Enlace de variables para Sistema de Riego CNC.</i>	38
<i>Ilustración 36 Configuración de las variables del sistema.</i>	38
<i>Ilustración 37 Sketch de programación de dispositivos.</i>	39
<i>Ilustración 38 Dashboard.</i>	39
<i>Ilustración 39 Definición de librerías.</i>	40

<i>Ilustración 40</i>	<i>Definición de pines</i>	<i>40</i>
<i>Ilustración 41</i>	<i>Configuración de la función “guardar_variables”</i>	<i>41</i>
<i>Ilustración 42</i>	<i>Configuración de la función “config_motores”</i>	<i>41</i>
<i>Ilustración 43</i>	<i>Configuración de la función “calibración”</i>	<i>42</i>
<i>Ilustración 44</i>	<i>Configuración de la función “regar”</i>	<i>43</i>
<i>Ilustración 45</i>	<i>Desplazamiento en X y Y</i>	<i>43</i>
<i>Ilustración 46</i>	<i>Cambios en el servidor</i>	<i>44</i>
<i>Ilustración 47</i>	<i>Archivo “thingProperties.h”</i>	<i>45</i>
<i>Ilustración 48</i>	<i>Interfaz de usuario funcional</i>	<i>46</i>
<i>Ilustración 49</i>	<i>Bandeja de germinadores utilizada para el proyecto</i>	<i>47</i>
<i>Ilustración 50</i>	<i>Mecanismo de riego Core XY</i>	<i>48</i>
<i>Ilustración 51</i>	<i>Almacenamiento de agua con acople de la bomba</i>	<i>49</i>
<i>Ilustración 52</i>	<i>Circuito electrónico sistema de riego por goteo aéreo</i>	<i>50</i>
<i>Ilustración 53</i>	<i>Inicio plataforma Arduino Cloud</i>	<i>59</i>
<i>Ilustración 54</i>	<i>Pestaña de acceso Arduino Cloud</i>	<i>60</i>
<i>Ilustración 55</i>	<i>Cuenta de acceso Arduino Cloud</i>	<i>60</i>
<i>Ilustración 56</i>	<i>Menú inicial Arduino Cloud</i>	<i>61</i>
<i>Ilustración 57</i>	<i>Ventana de configuración de dispositivos</i>	<i>62</i>
<i>Ilustración 58</i>	<i>Configuración de sensor y especificación de variable</i>	<i>63</i>
<i>Ilustración 59</i>	<i>Configuración de las variables</i>	<i>63</i>
<i>Ilustración 60</i>	<i>Ejemplo de diseño de Dashboard</i>	<i>64</i>

LISTA DE TABLAS

<i>Tabla 1 Herramientas y costos</i>	53
--	----

INTRODUCCIÓN

En la actualidad, la automatización de sistemas de riego trae grandes beneficios para los cultivos y un buen ahorro en insumos y demás elementos. Sin embargo, en el manejo de un sistema de riego es fundamental determinar el momento más adecuado para generar la aspersión, además de tener en cuenta otros factores, del estado de humedad del suelo o la planta y de la uniformidad en el reparto de agua del sistema.

A la hora de automatizar un sistema de riego se deben tener en cuenta todos los elementos que integran su sistema de control. Los componentes de los sistemas de control se pueden clasificar en cuatro grandes grupos: sensores y transductores (tensiómetros, manómetros, entre otros), actuadores (interruptores, electroválvulas, válvulas motorizadas, bombas, variadores de velocidad, y demás), acondicionadores de señal y unidades de control (ordenadores, programadores, entre otros).

Por tal motivo, en este proyecto se muestra el desarrollo del diseño CAD de la estructura de la planta, los elementos que le componen y su construcción, para tener todo un sistema de riego de goteo aéreo completo, implementando una Interfaz HMI (Interface Hombre Máquina), en la cual se prevé el envío de datos desde la plataforma de Arduino IoT Cloud (Internet of Things ó Internet de las cosas). Además de utilizar un mecanismo CORE X-Y (normalmente aplicado en el movimiento de CNCs).

OBJETIVOS

Objetivo general

Diseñar y construir un sistema de riego automatizado para germinadores que se encuentran dentro de invernaderos.

Objetivos específicos

- Identificar el sistema más adecuado de riego automático para germinadores dentro de invernaderos.
- Diseñar y construir el sistema mecánico para hacer el riego de las semillas.
- Diseñar y construir el sistema electrónico que controla la boquilla del aspersor de agua.
- Diseñar y construir la interfaz de usuario que permita manipular las condiciones de operación del sistema de riego.

MARCO CONCEPTUAL

1.1 Invernadero

Se entiende por invernadero un lugar cerrado, estático y accesible a pie, como se puede observar en la Ilustración 1, donde tiene de manera habitual una cubierta exterior translúcida de vidrio o de plástico, en la que se puede obtener un microclima con respecto al control de la temperatura, de la humedad y de otros factores ambientales, además, se pueden proporcionar sistemas automáticos de riego y ventilación, utilizado de gran manera para la producción y de cultivos de forma controlada [1].



Ilustración 1 Ejemplo de invernadero.

Fuente: [2]

1.2 Automatización

Se define automatización como el conjunto de procesos y técnicas que se aplican para sustituir al operario en las tareas tanto físicas como lógicas que requieren los procesos repetitivos. Generalmente, estas tareas están bien definidas y se pueden ejecutar como un conjunto de instrucciones simples. Teniendo en cuenta esta definición, se puede tratar a la automatización como la aplicación de la automática al control de procesos, generalmente industriales. Los procesos se entienden como toda forma de entrada,

transformación y salida de un sistema o elemento que se requiere modificar en alguna forma; dicha transformación está sujeta a perturbaciones del entorno. Los procesos industriales pueden clasificarse en tres grupos diferentes, los procesos continuos, los procesos discretos y los procesos tipo batch. Los procesos continuos se pueden comprender como toda salida de material de forma constante, como la purificación de agua o la generación eléctrica. Los procesos discretos tienen una salida de elementos de forma finita, como la fabricación de piezas o automóviles, y finalmente los procesos tipo batch son los que su salida se da por lotes, como la producción de refrescos o farmacéuticos. [3]

1.3 Germinador o semillero

Se llama germinador a la cámara de área acondicionada para la germinación de las semillas, que sirve como sembrador, garantizando su practicidad en el momento de sembrar semillas, que facilitan el brote de plántulas las cuales pueden ser trasplantadas, tienen forma de bandeja para huevos que permiten la preservación del calor y humedad y minimizar los aditivos químicos. [4]

La importancia del germinador en el proceso de la semilla es vital, pues si no hay germinación no hay planta, y sin planta no hay cosecha. El germinador permite el nacimiento precoz de las diferentes plantas a cultivar, el máximo rendimiento de la semilla y por ende plantas útiles, la obtención de mejores frutos y mayores cosechas, evitando el deshijamiento (eliminación de plántulas por exceso). En adición, es posible lograr una mayor protección contra las plagas, pues al no sembrar en suelo como se muestra en la Ilustración 2, se evita el problema producido por hongos, se logra también una adaptación de la plántula al medio donde se desarrollará, o bien organizar el semillero o germinador en el mismo sitio donde se hará el cultivo lo que hace posible la mecanización y hasta la robotización.

Por lo general los germinadores o semilleros se utilizan con un mínimo de profundidad de 5 cm y un diámetro que puede variar de 3 a 5 cm. El uso de germinadores es conveniente ya que la semilla puede permanecer en el pequeño contenedor entre 2-3 semanas y al sacar la plántula del germinador o pequeño contenedor no se lastima la raíz y puede efectuarse el trasplante con mayor facilidad, aun cuando el trasplante se realice a raíz desnuda ya que las raíces cuentan con su propio espacio y no se entrelazan con raíces de otra planta. [5]



Ilustración 2 Ejemplo gráfico de germinador o semillero.

Fuente: [5]

1.4 Sistema de riego.

Los sistemas de riego ofrecen una serie de ventajas que posibilitan racionalizar el agua disponible. Existen muchos y variados sistemas de riego, los cuales se encuentran en permanente revisión, ya que se trata de una tecnología joven que ha ido evolucionando al mismo tiempo que avanza la sociedad del bienestar. Las zonas verdes pasan de ser un lujo a ser una necesidad y el riego es la operación más importante para mantenerlas.

El sistema de riego se compone de primera mano de: bocas de riego, válvulas electroválvulas y llaves de estaciones; y segundo, de: aspersores, difusores, goteros y

exudantes; tercero: elementos destinados a distribuir el agua de acuerdo con una pluviometría predeterminada para los dispersores, difusores, bocas de riego, goteros, entre otros.

Los sistemas de riego ofrecen mayor exactitud y seguridad en el control de la instalación, operaciones mecanizadas exentas de errores, reducción de mano de obra, telemando y facilidad de programación y manejo, registro de datos para su análisis posterior.

Métodos de riego:

- Aspersión
- Goteo
- Exudante
- Informatizados

Como ejemplo gráfico, en la Ilustración 3, se muestra parte de las grandes ventajas de los métodos de riego, el cual es proporcionar una mejor gestión del agua, calidad en el riego, disminución de costos. Las aportaciones de agua pueden modificarse para que se adapten a las necesidades de las especies vegetales presentes, lo cual evita riegos inútiles o insuficientes que podrían ser dañinos para las plantas. [6] En general, el sistema garantiza una disminución global del consumo de agua entre un 30 y un 40%, independientemente del ahorro que supone el control del riego en cada zona concreta.

El riego se define como el sistema a partir del cual se logra un aporte de agua en los cultivos, capaz de suplir las necesidades hídricas de las plantas, que no se ven cubiertas únicamente por las precipitaciones. Este aporte puede realizarse con el fin de incrementar la producción de una parcela, transformando zonas agrícolas de secano en zonas de regadío [7].



Ilustración 3 Sistemas de riego.

Fuente: [2]

MARCO DE REFERENCIA

Los sistemas de riego automatizados son aquellos que proveen de agua a los cultivos de manera autónoma y son empleados generalmente en aquellos sistemas en que las variables deben tener mayor control. [8]

Estos sistemas manejan en su mayoría tipos de riego de aspersión o goteo, también existen otros que mezclan las dos técnicas, permitiendo la utilización de las ventajas que ambas representan, estas ventajas son:

2.1 Aspersión:

- Una de las principales ventajas es la necesidad de utilizar menor cantidad de material, en este caso tuberías y equipos y con ello, tener menor cantidad de gastos. En la mayoría de los sistemas de cobertura total, los aspersores se pueden espaciar de 6 a 18 metros de distancia y aun así proporcionar una aplicación uniforme. [9]
- Mayor versatilidad en el riego, ya que se puede cambiar de cultivo fácilmente.
- Tiempo de vida útil mayor, si se compara con los sistemas de goteo, En algunos campos, los productores pueden necesitar reemplazar secciones de sus tubos de goteo cada temporada. Esto se suma al costo de retirada y eliminación o reciclado. Las cifras no incluyen el costo adicional de protección contra heladas o sistemas de germinación. [9]

2.2 Goteo:

- Automatización, estos sistemas de goteo evitan el riego en zonas no objetivo, lo que conlleva a evitar emergencias y crecimiento de malezas. Además, evita la proliferación de enfermedades al reducir el contacto directo de las plantas con el agua.
- Adaptabilidad, tiene la facilidad de ser instalado en diferentes condiciones topográficas, además de manejar diferentes calidades y limitaciones salinas en el suelo.
- Alta eficiencia, estos sistemas utilizan solo la cantidad necesaria del agua, significando un ahorro en el uso de este recurso.

Algunos de estos sistemas automatizados de riego son los siguientes:

2.2.1 Aqua4D:

Es un sistema de riego inteligente por medio de la utilización de ondas electromagnéticas, lo cual traduce un aumento de los rendimientos en los diferentes cultivos, solución de la salinidad la cual es muy importante en las plantas para su correcto desarrollo y a su vez un ahorro significativo en la utilización de recursos. [10]

Este sistema posee una eficiencia hídrica gracias a su efecto sobre las moléculas de agua, el suelo logra mantenerse húmedo por más tiempo, lo que supone para un agricultor un ahorro medio de agua de un 25% a 30%, hasta más cuando es combinado con diferentes tecnologías Ag 4.0, por lo tanto, su innovación disruptiva, supone un nuevo paradigma de gestión sostenible del agua, disminuyendo la utilización de recursos y aumentando el rendimiento. [10]

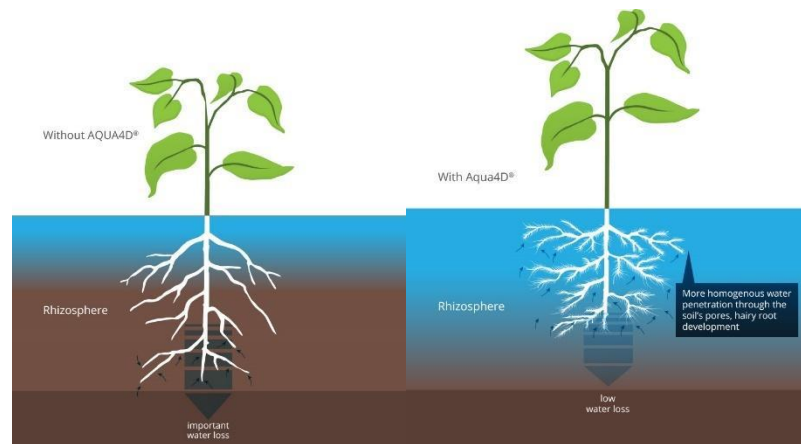


Ilustración 4 Diferencias en cultivo con la implementación de Aqua4D.

Fuente: [10]

En cuanto a la Ilustración 4, en la sección derecha se puede observar que este sistema disuelve eficazmente las sales y minerales presentes en el agua, aumentando el rendimiento de estos cultivos, teniendo como gran ventaja el proceso de fertilizantes y su eficacia.

2.2.2 Riego por nebulización:

Este es un sistema de riego el cual expulsa agua en forma de neblina, a través de emisores colocados en las partes superiores de los cultivos, el cual además de su función de distribuir agua y fertilizantes, ayuda a disminuir la temperatura dentro del invernadero a su vez que aumenta su humedad. [11]

Estos emisores son llamados nebulizadores ya que expulsan el agua en forma de niebla fina, (Ver Ilustración 5), allí el agua sale a presión por medio de un orificio de diámetro pequeño, de forma que el chorro saliente, golpea contra una pared cóncava que lo despidе y distribuye de forma nebulizada. Estos cultivos generalmente funcionan con presiones elevadas. [11]

Este riego se compone de:

- Sistema de bombeo
- Sistema de filtración
- Sistema de aplicación de fertilizantes o fungicidas
- Red de tuberías

Este sistema tiene grandes ventajas como: no compactar el suelo, manteniéndolo suelto para una mejor aireación y distribución de la humedad de este, no ocasionan daños en los cultivos dependiendo de su etapa y tipo de cultivo.



Ilustración 5 Riego por nebulización.

Fuente: [11]

Su única desventaja es su implementación en grandes cuartos de cultivo, debido a que aumentaría considerablemente su costo en todos los sentidos.

2.2.3 Two Valve:

En Chile, un estudiante de Ingeniería Civil Mecánica desarrolló un sistema innovador, este tiene como base la robótica modular, que permite a los agricultores aplicar la automatización a los sistemas de riego de forma fácil, sencilla y sin intermediarios. [12]

Este es un sistema creado y pensado para zonas de difícil acceso, con una alta salinidad en el ambiente debido a su incremento de concentración en la tierra y la gran cantidad de radiación, lo cuales dificultan el uso de tecnologías, también debido a su difícil acceso elevan el costo de mantenimiento, transporte, entre otros.

A diferencia de los sistemas convencionales, que ocupan más de 40 válvulas, en los que, si una de ellas presenta inconvenientes, la línea para por completo hasta que el técnico a cargo pueda hacer la respectiva revisión. [12] Dichos sistemas, tienen la ventaja de implementar unidades de riego autónomo, como se puede detallar en la Ilustración 6, permitiendo que, al presentarse alguna falla, pueda ser atendida de manera inmediata, sin ocupar algún intermediario.



Ilustración 6 Sistemas Two Valve.

Fuente: [12]

MARCO NORMATIVO

- **Resolución 000132 de 2021 (Ministerio de Agricultura y Desarrollo Rural):**
Por la cual se adopta el Manual de Normas Técnicas para sistemas de riego y drenaje a nivel predial.
- **Ministerio de Agricultura, 2015:** Manual de normas técnicas básicas para la realización de proyectos de adecuación de tierras.
- **Decreto No 182 del 4 de marzo de 1968, pág. 497:** por el cual se reglamenta el uso de aguas, operación, conservación y mantenimiento de los distritos de riego y avenamiento.

SISTEMA DE RIEGO DE GOTEO AÉREO

4.1 Sistemas de riego aéreo

Son sistemas de riego en los que el agua se aplica a los cultivos en forma de lluvia, mojando la totalidad de la planta, así como la superficie cultivada.

Tiene como desventaja, que el agua no es aprovechada como en otros sistemas de riego localizado, sin embargo, un factor de suma importancia en la selección del tipo de riego aéreo será el formato de la especie cultivada, hecho que debe conducir a un diseño adecuado que permita obtener una menor o mayor precipitación (l/hm^2), para lograr aumentar la eficiencia en la penetración de las gotas hasta llegar al suelo. [13]

4.1.1 Sistema de Riego Aéreo por Goteo.

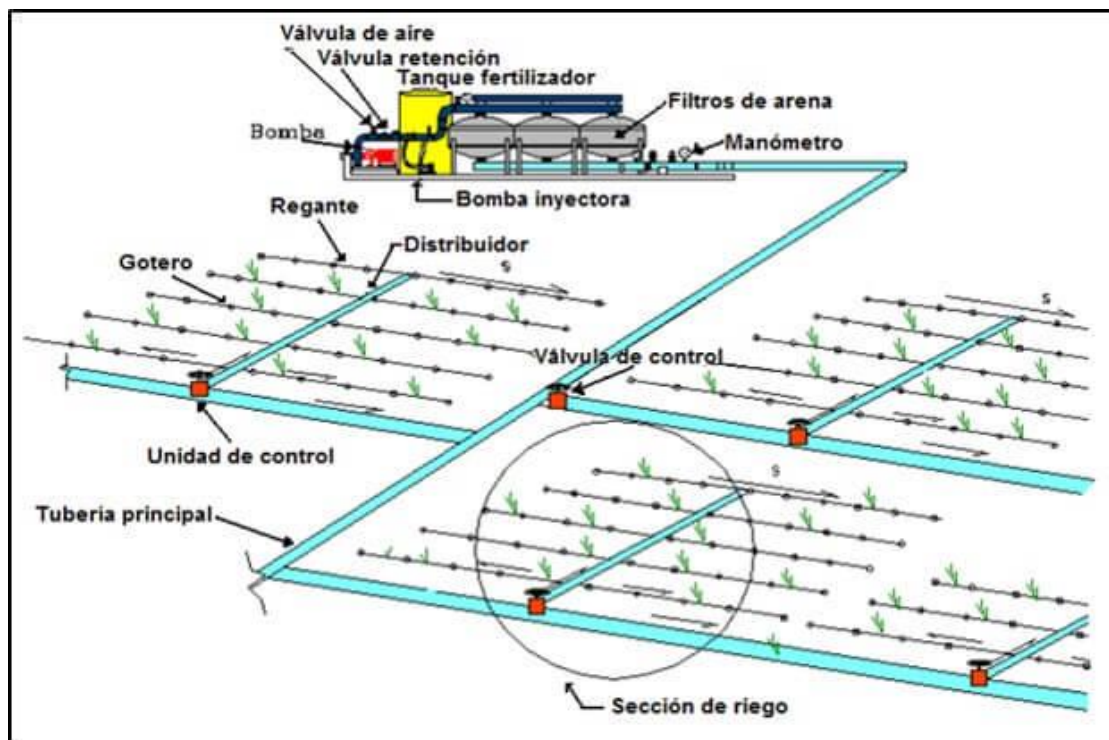


Ilustración 7 Componentes principales de un sistema de riego por goteo.

Fuente: [13]

Como se aprecia en la Ilustración 7, este sistema entrega el agua y fertilizantes directamente a la zona radicular del cultivo, en la cantidad correcta y en el momento adecuado, por lo tanto, cada planta recibe exactamente lo que necesita, cuando lo necesita para desarrollarse óptimamente. Gracias al riego por goteo, los productores pueden tener mejores rendimientos mientras ahorran agua, así como fertilizantes, energía e incluso agroquímicos.

¿Cómo funciona?

El agua y los nutrientes se entregan en el campo a través de un sistema de tuberías llamados “líneas de goteo”, (visibles en la Ilustración 7), las cuales contienen pequeños dispositivos llamados “goteros”. Cada gotero emite un flujo controlado de gotas que contienen agua y fertilizantes, resultando en una uniforme aplicación de agua y nutrientes directamente a la zona radicular de cada planta a lo largo de todo el campo. [15]

4.2 Requerimientos.

La clave para un buen diseño de sistema de riego consiste en establecer los parámetros que determinan las restricciones a que debe someterse el proyecto, como topografía del terreno, tipo de suelo, localización, cantidad y calidad del agua disponible, tipo y ubicación de la alimentación, entre otros. [14]

Normalmente este proceso consta de dos fases:

- El diseño agronómico consiste en garantizar que el sistema de riego sea capaz de suministrar, con una alta eficiencia de aplicación las necesidades hídricas del cultivo durante el período de máximo consumo, consiguiendo también mojar el volumen de suelo suficiente (bulbo húmedo) para su adecuado desarrollo y un efectivo control de sales. La finalidad de esta fase es obtener los siguientes parámetros:

- (i) Necesidades totales de riego;
 - (ii) Caudal y número de emisores por planta o por unidad de superficie;
 - (iii) Tiempo de aplicación.
- La capacidad del sistema de riego debe ser suficiente para satisfacer las necesidades del cultivo en la etapa de máxima demanda de agua, con base al tipo de cultivo y a las variables climáticas. También esto permite determinar el volumen mínimo de suelo que ha de mojarse para garantizar una adecuada aportación de agua y nutrientes. [14]

4.3 Equipos.

4.3.1 Bomba para sistemas de goteo.



Ilustración 8 Bomba para el sistema de riego aéreo por goteo
Fuente: Autor.

La Ilustración 8 muestra una bomba para el sistema de riego, funciona a 12V y normalmente es utilizada en vehículos para la parte del limpiaparabrisas. Se utiliza mediante PWM para regular la salida del agua.

4.3.2 Manguera de 6mm



Ilustración 9 Manguera neumática de 6mm.

Fuente: [16]

La Ilustración 9, muestra una manguera neumática de 6mm de diámetro, la cual va conectada al deslizador central (ver Ilustración 17), además lleva unos conectores de acople rápido P601 que se ven a continuación y uno de los conectores va hacia la boquilla de aspersión y de allí a la bandeja del germinador.



Ilustración 10 Acople rápido PC6 01.

Fuente: [16]

DISEÑO MECÁNICO

Para el diseño, se tuvo muy en cuenta el mecanismo de una CNC, realizando un prototipo a escala, aplicando los componentes para el riego en goteo, y dirigido hacia las cubetas con las semillas, con el fin de lograr una germinación más controlada y eficiente.

5.1 ¿Qué es una CNC?

El Control Numérico por Computadora, es un sistema que permite controlar en todo momento la posición de un elemento físico. Su funcionamiento está basado en el posicionamiento sobre los ejes X, Y, Z. Gracias a lo cual, a una misma pieza se la puede taladrar, cortar, roscar, fresar, o desbastar en todos sus planos de forma totalmente automática. [16]

Cabe resaltar que, aunque el sistema CNC es muy conocido en el mundo del mecanizado, no lo es menos en otras aplicaciones como el corte por láser, chorro de agua, aire comprimido o electroerosión, debido a que están basados en el mismo principio de funcionamiento, con la diferencia del sistema de corte empleado.

5.2 CORE [X, Y]

Es un sistema cinemático normalmente utilizado en impresoras 3D y otras aplicaciones de control de movimiento. Está diseñado para proporcionar un movimiento preciso y eficiente del cabezal de impresión o la herramienta en un plano bidimensional. El mecanismo CORE XY es conocido por su capacidad para eliminar la necesidad de motores pesados y voluminosos en el carro en movimiento, lo que da como resultado un movimiento más suave y preciso. [18]

5.2.1 Principio de operación:

En un sistema CORE XY, el cabezal de impresión o la herramienta están soportados por dos correas que están dispuestas en una configuración entrecruzada o diagonal. Las correas están unidas a poleas o motores ubicados en posiciones fijas fuera del área de la cama de impresión. La disposición entrecruzada de las correas crea una serie de

paralelogramos que forman la estructura central del mecanismo, así como se puede observar en el siguiente esquema gráfico:

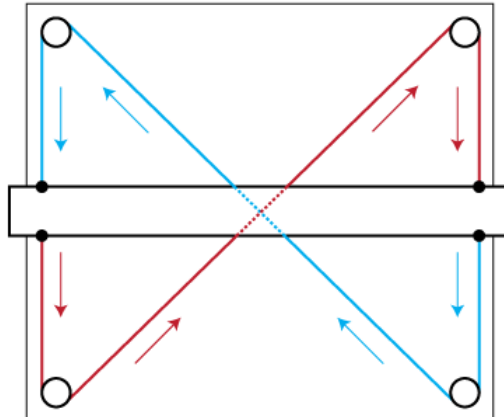


Ilustración 11 Disposición de las correas.

Fuente: [18]

Para mover el cabezal de impresión o la herramienta en las direcciones X e Y, las correas son accionadas por poleas o motores (Ver la Ilustración 12). Al controlar el movimiento de las dos correas de forma independiente, el cabezal de impresión se puede colocar con precisión en cualquier ubicación deseada dentro del plano XY. [18] Las correas suelen estar hechas de un material resistente y flexible, como caucho reforzado o fibra de vidrio, lo que garantiza durabilidad y una transmisión de movimiento precisa.

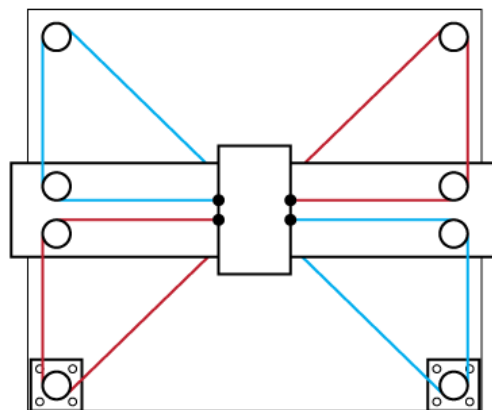


Ilustración 12 Accionamiento de las correas.

Fuente: [18]

Sin embargo, al querer diseñar un mecanismo Core XY, se deben considerar ciertas ecuaciones de movimiento y para ello se considera el siguiente como referencia:

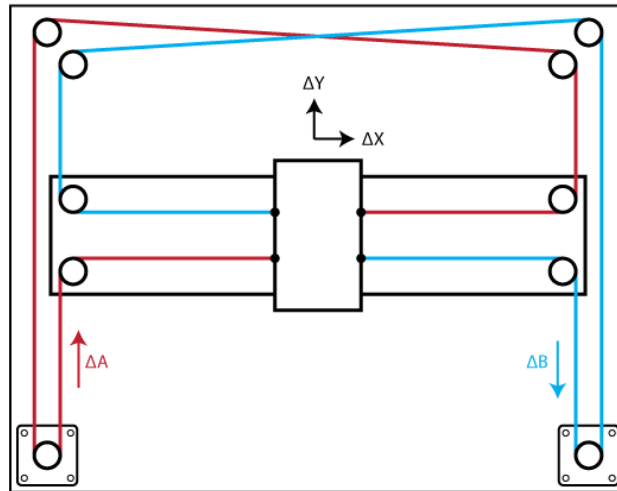


Ilustración 13 Mecanismo Core XY de referencia.

Fuente: [18]

Donde, las ecuaciones de movimiento se definen por:

- $\Delta X = \frac{1}{2}(\Delta A + \Delta B)$
- $\Delta Y = \frac{1}{2}(\Delta A - \Delta B)$
- $\Delta A = (\Delta X + \Delta Y)$
- $\Delta B = (\Delta X - \Delta Y)$

Ahora, sumado a las ecuaciones de movimiento, y como es un sistema accionado por correa, se debe tener en cuenta para su calibración los pasos por milímetro, así que se consideran los siguientes datos:

$s =$ Pasos por revolución.

$x =$ Steps/mm.

$p =$ Pitch de correa.

$t =$ Dientes de polea.

$m =$ Micro pasos.

$$x = \frac{s * m}{p * t}$$

Ecuación 1. Pasos por milímetro.

Y para calcular los milímetros por paso:

$$y = \frac{p * t}{s * m}$$

Ecuación 2. Milímetros por paso.

Teniendo las fórmulas planteadas, se procede a realizar el reemplazo de los valores con respecto al número de pasos del motor utilizado para el mecanismo.

Datos:

$$s = 200 \text{ steps}$$

$$x = \text{Steps/mm.}$$

$$p = 2\text{mm}$$

$$t = 20$$

$$m = 1$$

$$x = \frac{(200 \text{ steps}) * (1)}{(2\text{mm}) * (20)}$$
$$x = 5 \text{ steps/mm}$$

$$y = \frac{(2\text{mm}) * (20)}{(200 \text{ steps}) * (1)}$$
$$y = 0,2 \text{ mm/steps}$$

5.3 Diseño CAD de la CNC para el mecanismo de riego.

Teniendo ya unos conceptos claros y continuando con el diseño mecánico, la CNC, que se puede observar en la *Ilustración 14* consta de 2 ejes: X y Y, por los cuales se desplaza el mecanismo de riego. Al tener 2 ejes, se requieren de igual cantidad motores, así como de poleas y correas, éstas siendo de ayuda en desplazamiento.

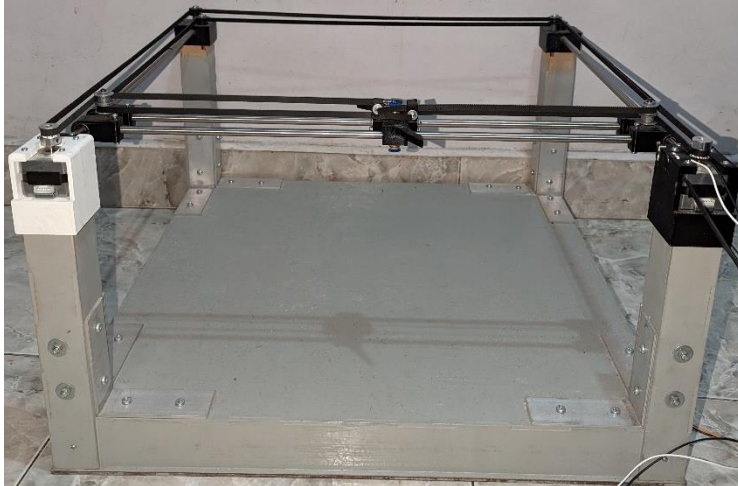


Ilustración 14 Estructura de CNC para sistema de riego por goteo aéreo.
Fuente: Autor.

Las dos correas conectadas en el centro ayudan a la dirección de giro de los motores. En laterales, se pueden observar dos deslizadores que permiten un movimiento de arriba abajo (eje y) y en posición superior, se encuentra un deslizador que permite el movimiento de derecha a izquierda (eje x). Este mecanismo permite que, a través de unas coordenadas asignadas, el riego se pueda operar de manera uniforme y guiada.

El sistema cuenta también con dos finales de carrera, que permiten tener una posición inicial (0,0).

Nota: Todos los deslizadores tienen rodamientos lineales, y se describen de manera detallada a continuación, con respecto al plano explosionado que se muestra en la *Ilustración 15*.

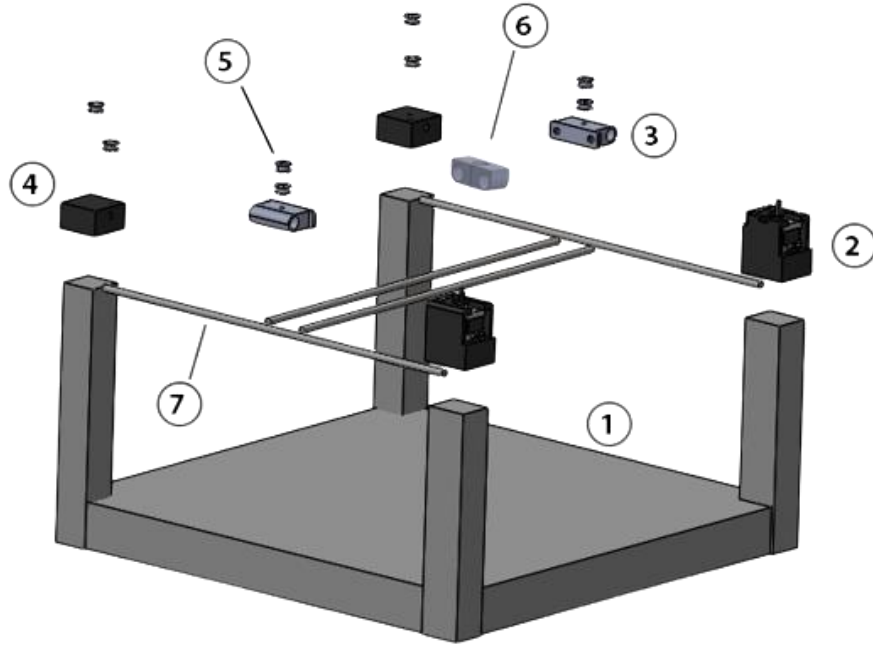


Ilustración 15 Plano explosionado en Software CAD.

Fuente: Autor.

A continuación, se explicarán cada uno de los numerales del plano explosionado de la Ilustración 15.

1. Base de la CNC:

Esta base echa en madera con grosor de 3mm y cortada a láser, cuenta con 4 pilares que sirven de soporte a los demás dispositivos. Dichos pilares tienen un refuerzo con placas de aluminio, en forma de L, conservando un ángulo de 90°, esto con el fin de brindar mayor soporte y conservar la tensión en las poleas.

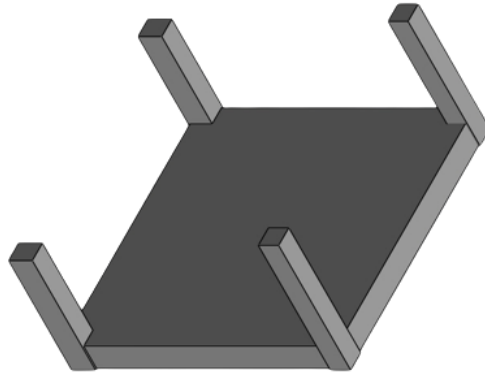


Ilustración 16 Base de la CNC.

Fuente: Autor.

2. Cubierta de los motores

Las carcasas de los motores paso a paso (ver Ilustración 17), cumplen la función de armadura para los mismos, los cuales accionan las poleas para generar el movimiento de los desplazadores que están incorporados a los rieles externos e internos. Estas carcasas cuentan con una pieza adicional, que tiene como prioridad sostener las varillas de los deslizadores a través de tornillos.

Cabe resaltar, que estas piezas están impresas en 3D, utilizando PLA como material.

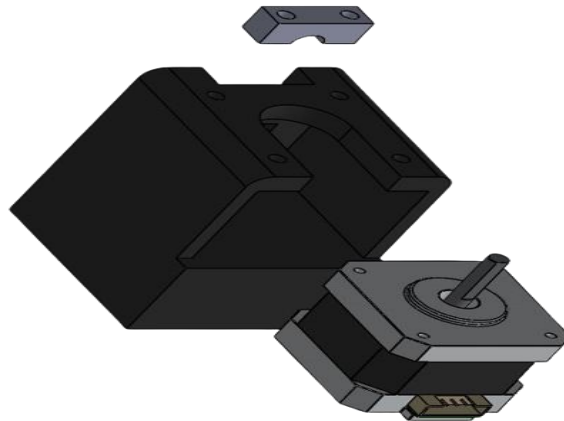


Ilustración 17 Cubierta de motores.

Fuente: Autor.

3. Deslizador lateral

Tiene rodamientos lineales, los cuales permiten el movimiento en el eje X. También sirven de soporte para el deslizador central.

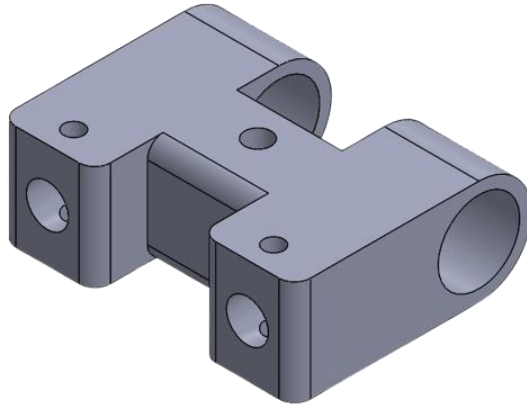


Ilustración 18 Deslizador lateral

Fuente: Autor.

4. Cubierta de poleas.

Estas piezas sirven de acople en el pilar, y dan soporte a cada una de las poleas.

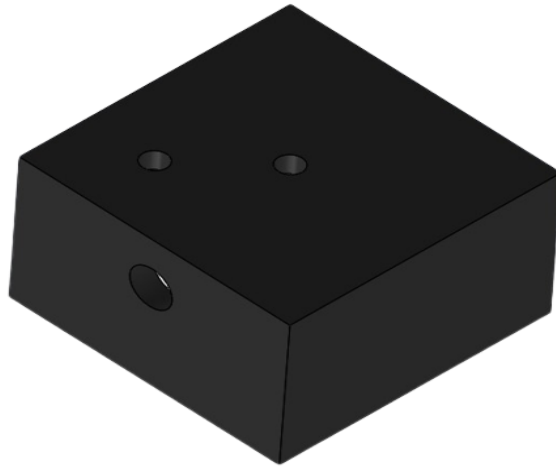


Ilustración 19 Cubierta de poleas.

Fuente: Autor.

5. Polea GT2.

Polea específicamente utilizada para correas GT2 de 6mm, e impresa en PLA.

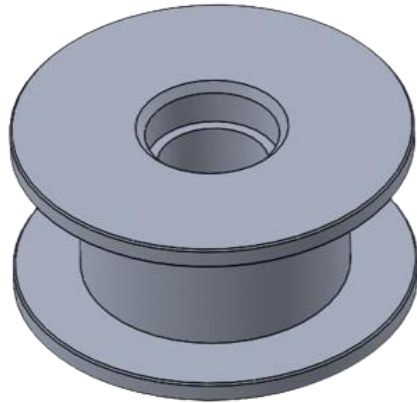


Ilustración 20 Polea impresa GT2.

Fuente: Autor.

6. Deslizador central.

Está enlazado a las varillas que sostienen los deslizadores laterales. Cuenta con dos soportes y es parte fundamental para el mecanismo de riego. Sostiene las correas de ambos motores para poder realizar el movimiento. Y en el orificio central, van dos acoples en lo que se introduce la manguera para el riego.

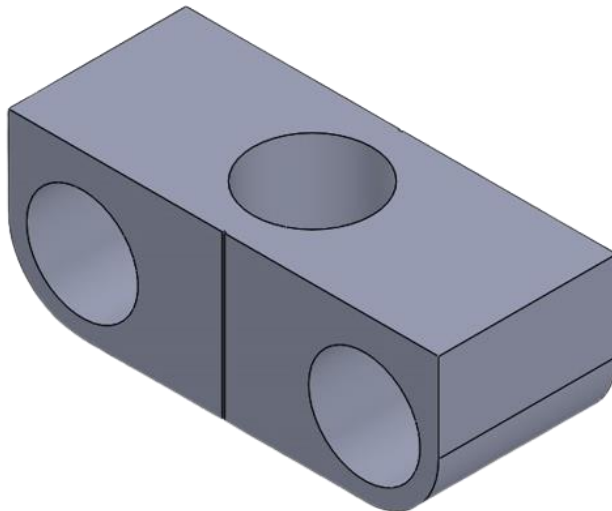


Ilustración 21 Deslizador central.

Fuente: Autor.

7. Varilla de deslizador.

Conecta a los dos pilares y a los dos deslizadores centrales, esta permite el correcto funcionamiento del sistema.



Ilustración 22 Varilla de deslizador.

Fuente: Autor.

DISEÑO ELECTRÓNICO

7.1 Componentes electrónicos

7.1.1 Módulo ESP32:

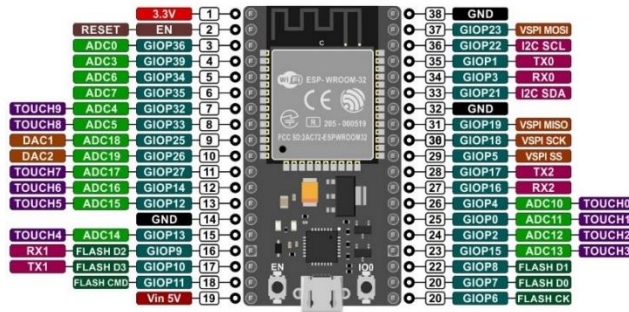


Ilustración 23 Módulo Wifi ESP32.

Fuente: [19]

El ESP32 es una tarjeta de desarrollo, potente, versátil y ampliamente utilizado que integra conectividad Wi-Fi y Bluetooth. Además, incluye una amplia gama de interfaces periféricas como pines GPIO (entrada/salida de propósito general), UART (receptor-transmisor asíncrono universal), SPI (interfaz periférica en serie), I2C (puerto y protocolo de comunicación serial) y otros más, que se pueden apreciar en la ilustración anterior.

Esto lo hace adecuado para diversas aplicaciones, incluidos dispositivos de Internet de las cosas (IoT), sistemas de automatización del hogar, robótica, tecnología portátil y automatización industrial.

7.1.2 Motor paso a paso 17HS3401 y Driver A4988:

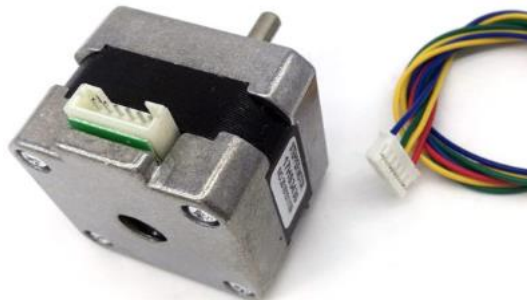


Ilustración 24 Motor Nema 17HS3401.

Fuente: [20]

El motor NEMA 17 17HS3401, mostrado en la Ilustración 24, se usa ampliamente en diversas aplicaciones, particularmente en el campo de las impresoras 3D, las máquinas CNC y la robótica. Se elige por varias razones:

- Tecnología de motor paso a paso: Los motores paso a paso son conocidos por su control preciso de la posición y la velocidad. Ofrecen un alto par a bajas velocidades y pueden moverse en pequeños pasos incrementales, lo cual es esencial para aplicaciones que requieren un posicionamiento preciso, como impresoras 3D y máquinas CNC.
- Par de retención: el motor 17HS3401 tiene un par de retención especificado, que es la cantidad de par que el motor puede ejercer para mantener una posición estacionaria cuando está energizado, pero no gira. El par de retención es una consideración importante en aplicaciones que requieren mantener la posición sin bloqueo mecánico externo.

Para su óptimo funcionamiento, se requiere de la implementación de un controlador, esto permite amplificar la corriente y su función radica en tomar una señal de control de baja corriente y luego convertirla en una señal de corriente más alta que pueda conducir un motor. Para fines del proyecto, se optó por usar el Driver A4988 que se verá a continuación:



Ilustración 25 Driver A4988, para motor PAP.

Fuente: [20]

7.1.3 Transistor FDD8580 SMD D-pack:

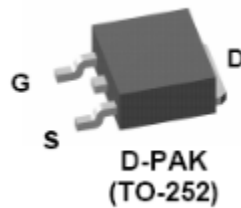


Ilustración 26 Transistor FDD8580 SMD D-pack

Fuente: [21]

También conocido como MOSFET de potencia, se considera el más adecuado por las siguientes razones:

- Manejo de alta potencia: el transistor FDD8580 está diseñado para manejar niveles relativamente altos de potencia.
- Baja resistencia de encendido: el transistor FDD8580 exhibe una baja resistencia de encendido, lo que significa que tiene una resistencia mínima cuando conduce corriente en su estado activo. Esta baja resistencia de encendido reduce la pérdida de energía y mejora la eficiencia en las aplicaciones de energía.
- Velocidad de conmutación rápida: el transistor FDD8580 presenta una velocidad de conmutación rápida, lo que le permite cambiar rápidamente entre los estados de encendido y apagado. Esta característica es esencial en aplicaciones que requieren tiempos de respuesta rápidos y control preciso, como control de motores, convertidores de potencia y fuentes de alimentación conmutadas.

7.1.4 Regulador 5v LDO LD1117 SMD:

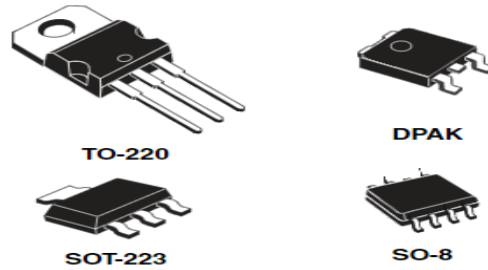


Ilustración 27 Regulador LD1117.

Fuente: [22]

El LD1117, visto en sus diferentes presentaciones en la anterior ilustración, es en pocas palabras un circuito integrado regulador de voltaje (IC) que se usa comúnmente para regular y proporcionar un voltaje de salida estable en circuitos electrónicos.

Está diseñado para aceptar un voltaje de entrada más alto y regularlo hasta el voltaje de salida deseado con un voltaje de caída bajo. El voltaje de caída se refiere a la diferencia de voltaje mínima requerida entre la entrada y la salida para que el regulador funcione correctamente.

7.1.5 Finales de carrera:



Ilustración 28 Final de carrera para CNC.

Fuente: [20]

Los finales de carrera son componentes vitales para brindar seguridad en la operación y, además, tienen las siguientes ventajas:

- Seguridad de la máquina: pueden detectar los límites del movimiento de la máquina en cada eje, evitando el exceso de recorrido y posibles colisiones con obstrucciones físicas.
- Posicionamiento inicial: los finales de carrera son normalmente utilizados para establecer una posición inicial o de referencia constante y precisa para la máquina CNC. Al ubicar con precisión la máquina en un punto de inicio conocido, garantiza una operación óptima.
- Detección de fin de recorrido: ayudan a identificar los puntos finales o límites del recorrido de la máquina en cada eje. Cuando la máquina alcanza estos límites, los interruptores de límite se activan y le indican al controlador que detenga el movimiento. Esto evita que la máquina intente moverse más allá de sus límites mecánicos, evitando daños al mecanismo.
- Posicionamiento: esto mejora la precisión y la eficiencia a modo general de las operaciones del sistema.
- Detección de errores y diagnóstico de fallas: pueden usarse para detección de errores o fallas en el sistema.

7.1.6 Fuente 12V 5A:



Ilustración 29 Adaptador de 12v 5A.

Fuente: Autor.

Para energizar el circuito del sistema de riego, se optó por un adaptador de 12 voltios de salida y 5 amperios. Este componente es esencial para suministrar la energía eléctrica necesaria para los diferentes dispositivos dentro de la planta.

7.2 Esquema de conexiones

Con respecto a los parámetros mencionados anteriormente en el capítulo de Diseño mecánico y en conjunto con los componentes electrónicos seleccionados para el funcionamiento del sistema de riego, se plantea el siguiente esquema de conexiones.

Este diseño que se puede observar a continuación fue realizado en el software de simulación y diseño electrónico: Fritzing.

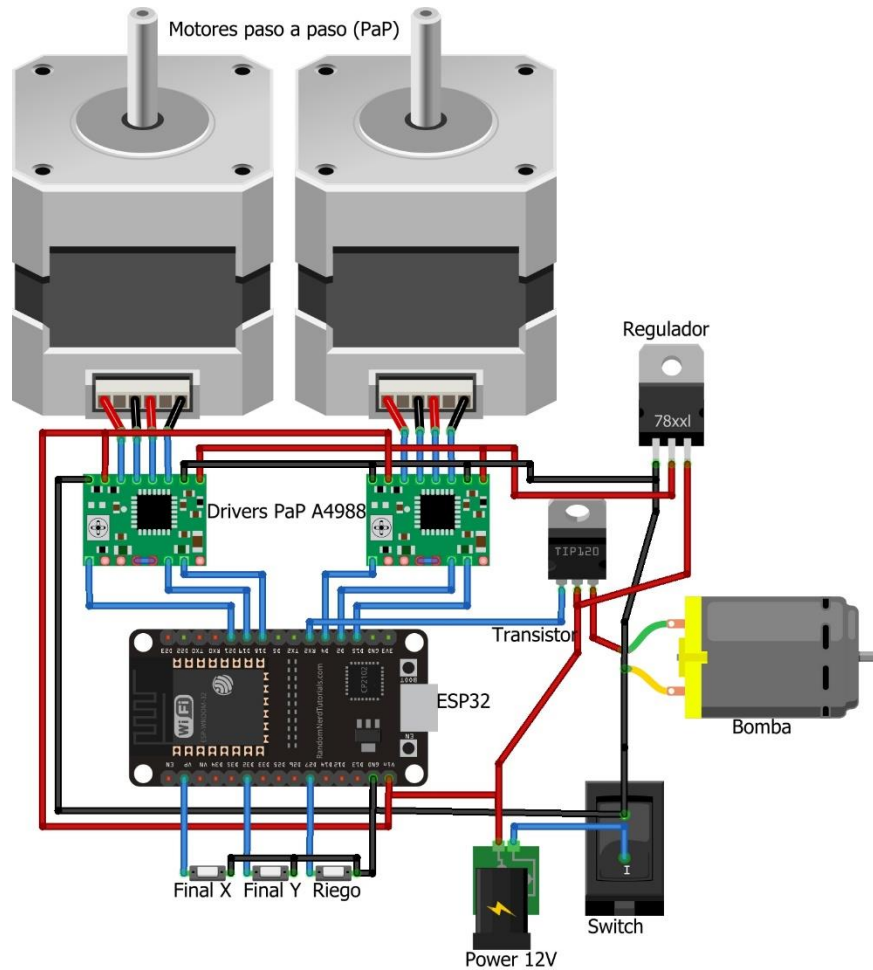


Ilustración 30 Esquema de conexiones circuito eléctrico de Sistema de Riego.

7.2.1 Diseño PCB

Para el diseño de la placa PCB, teniendo en cuenta las conexiones anteriores, se utilizó el software Eagle CAD.

En Eagle CAD se pueden diseñar placas de circuito impreso (PCB). Tiene la ventaja de poder diseñar y editar esquemas de circuitos utilizando un editor de esquemas. Ofrece una amplia gama de componentes electrónicos y símbolos que se pueden colocar e interconectar en el lienzo esquemático para representar el circuito deseado.

Una vez que se completa el diseño esquemático, se puede transferir el diseño al editor de diseño de PCB. En el editor de diseño, pueden organizar los componentes en la PCB y definir las conexiones físicas entre ellos. Este software, proporciona funciones para enrutar trazas, agregar vías, definir vertidos de cobre y ajustar el diseño para cumplir con los requisitos de fabricación específicos, por ello, fue el más apto para realizar el diseño de la PCB del circuito del sistema de riego y quedó de la siguiente manera, como se muestran en las Ilustraciones 31 y 31.

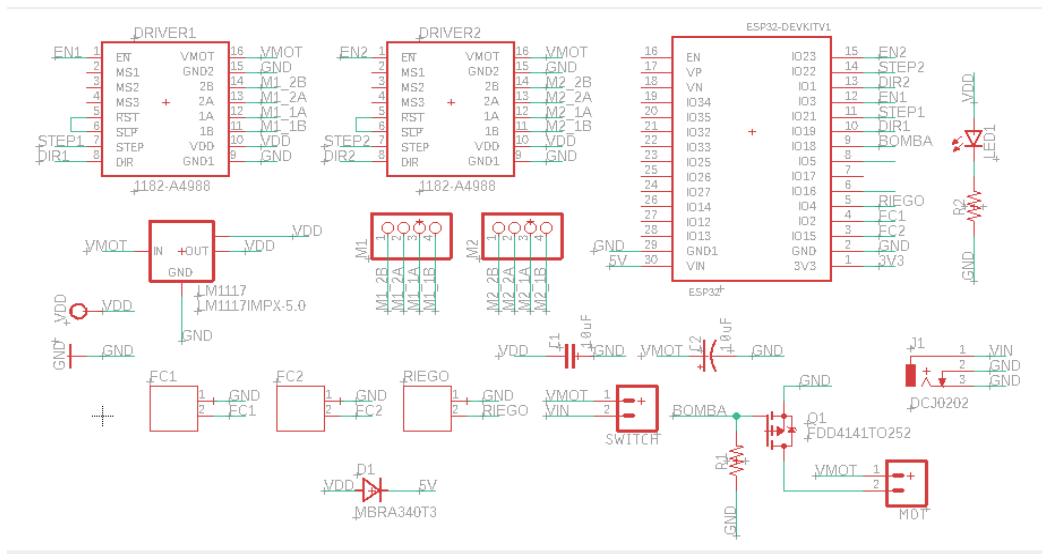


Ilustración 31 Trazo del esquema de conexiones en Eagle.

Fuente: Autor.

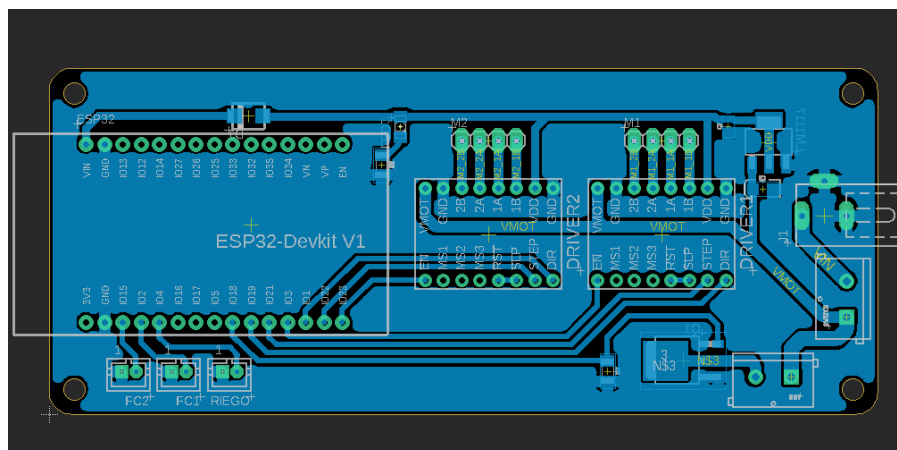


Ilustración 32 Layout de la PCB.

Fuente: Autor.

DESARROLLO DEL SOFTWARE PARA EL CONTROL DEL SISTEMA DE RIEGO

8.1 Interfaz de usuario

8.1.1 Arduino IoT Cloud

Actualmente, Arduino aparte de sus tarjetas programables, ofrece un gran servicio que es el Arduino IoT Cloud. Esta herramienta permite tener un control óptimo del sistema a distancia, además de guardar toda la información recopilada en la nube y permitir hacer un análisis del comportamiento de la planta con la que se esté trabajando.

La plataforma Arduino IoT Cloud (que se puede observar en la Ilustración 30) puede ser programada con bocetos, generando una interfaz amigable y diversa. Además, tiene la ventaja de tener conectividad con dispositivos o hardware de terceros, lo que lo hace amigable con otros microcontroladores o tarjetas programables. [17]

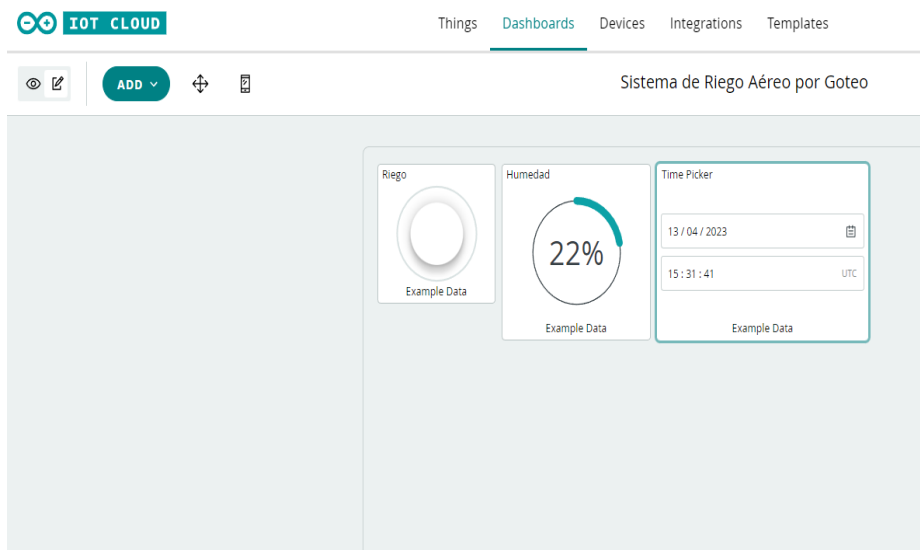


Ilustración 33 Ejemplo de la interfaz de usuario organizada en el Dashboard de Arduino IoT Cloud.

Fuente: Arduino IoT Cloud.

Nota: En la sección de anexos, al finalizar el documento, podrá encontrar una Guía básica para el ingreso y configuración de la Plataforma de Arduino IoT Cloud.

Lo esencial para este proceso fue adquirir una suscripción “Maker” por valor de \$6.99 dólares, Arduino Cloud de una duración de un mes, para aprovechar los diferentes beneficios que brinda la plataforma y poder utilizar en totalidad los sensores dispuestos para el trabajo.

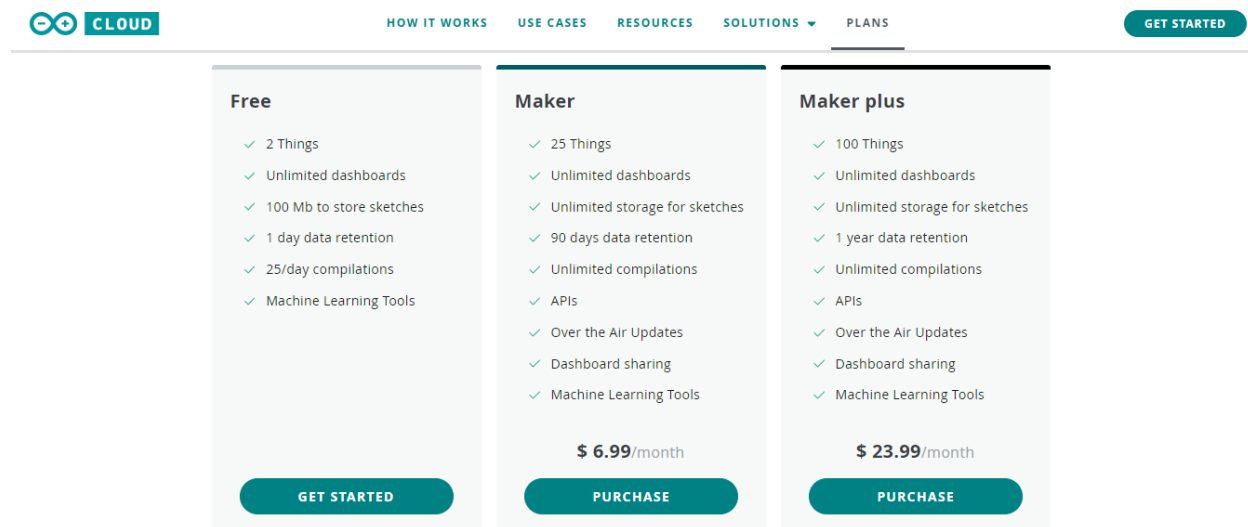


Ilustración 34 Suscripción Arduino Cloud.

Fuente: Arduino Cloud.

En la ilustración anterior, se pueden ver categorizadas tres suscripciones: Free, Maker y Maker Plus, la gran diferencia radica no sólo en valor si no en los dispositivos y variables que pueden ser conectados, además de la información que puede ser almacenada. Para casos prácticos del trabajo, se optó por la suscripción Maker, ya que tiene grandes ventajas como:

- 3 meses de retención de datos en la nube.
- Variables ilimitadas por cada sensor o tarjeta de adquisición de datos.
- Capacidad para conectar hasta 25 dispositivos.
- Transmitir los datos de los sensores a un dispositivo móvil, incluso cuando la aplicación no está en uso.

Después de tener lista la suscripción, se procede a configurar en la sección “Things”, (ver Ilustración 31), las estaciones de trabajo, microcontroladores o tarjetas de adquisición de datos y las variables de cada uno de los proyectos.

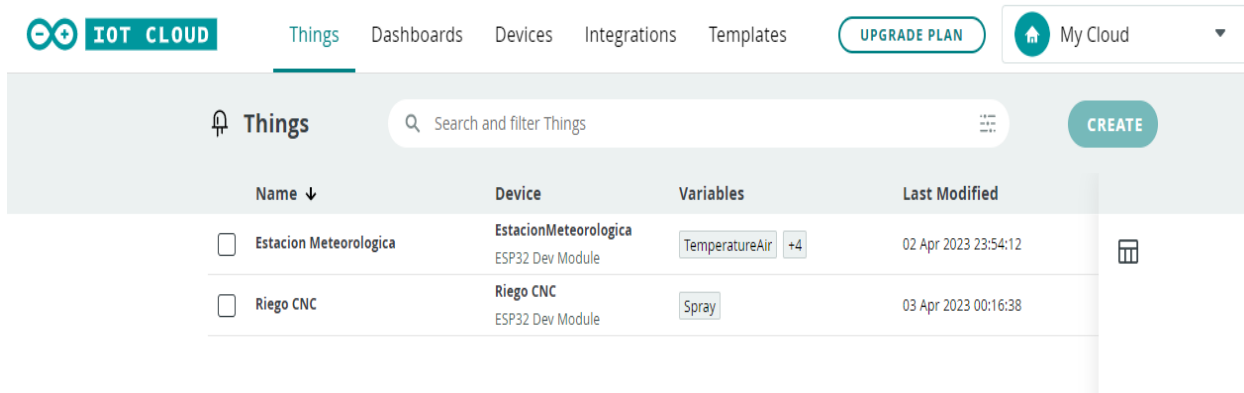


Ilustración 35 Enlace de variables para Sistema de Riego CNC.
Fuente: Arduino Cloud.

Al hacer clic en la estación de trabajo, “Riego CNC”, se despliegan todas las opciones de configuración, en especial de las variables, para determinar si son tipo entero o flotante, como se observa en la siguiente Ilustración 32, y además se crea el enlace con el dispositivo de control.

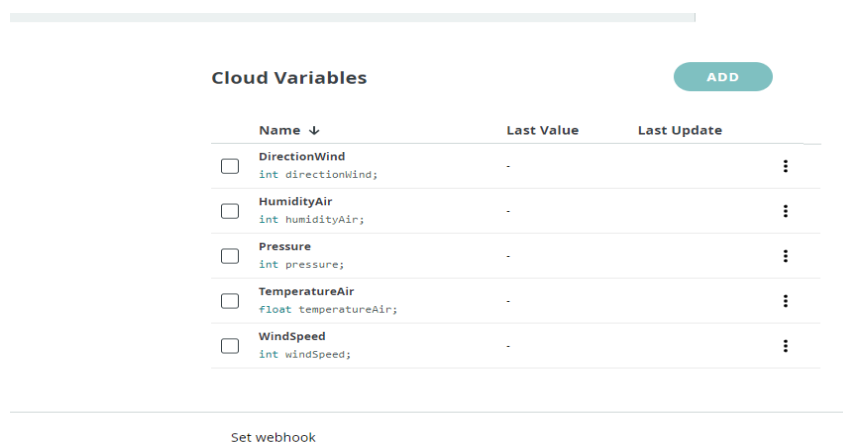


Ilustración 36 Configuración de las variables del sistema.
Fuente: Arduino Cloud.

En la sección de “sketch”, que se observa a continuación, se realiza toda la parte de codificación, cabe destacar que parte de las ventajas de esta plataforma de Arduino Cloud, es que trae por determinado parte del código para tener una base de programa y facilita de esta manera la lista de instrucciones para cada una de las variables.

```
16 which are called when their values are changed from the Dashboard.
17 These functions are generated with the Thing and added at the end of this sketch.
18 */
19
20 #include "thingProperties.h"
21
22 void setup() {
23   // Initialize serial and wait for port to open:
24   Serial.begin(9600);
25   // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
26   delay(1500);
27
28   // Defined in thingProperties.h
29   initProperties();
30
31   // Connect to Arduino IoT Cloud
32   ArduinoCloud.begin(ArduinoIoTPreferredConnection);
33
34   /*
35   The following function allows you to obtain more information
36   related to the state of network and IoT Cloud connection and errors
37   --
```

Ilustración 37 Sketch de programación de dispositivos.

Fuente: Arduino Cloud.

Nota: Dentro del servidor siempre queda guardado el sketch, y se puede subir de manera remota vía OTA. Para fines de proyecto y en caso tal de que algún inconveniente surja por cambio, se dejará una copia del código original en un enlace directo en Drive.

Después de determinar la programación para el sistema de riego, se procede a esquematizar la interfaz para trabajar no sólo desde un computador sino también a través de un dispositivo móvil. Esto dentro de la plataforma de Arduino Cloud se denomina “Dashboard”, y allí se hace la recepción de la información adquirida a través de los sensores y se puede hacer una supervisión de la estación en tiempo real. A continuación, se muestra un ejemplo de la construcción de la interfaz y su visualización para ambos dispositivos.

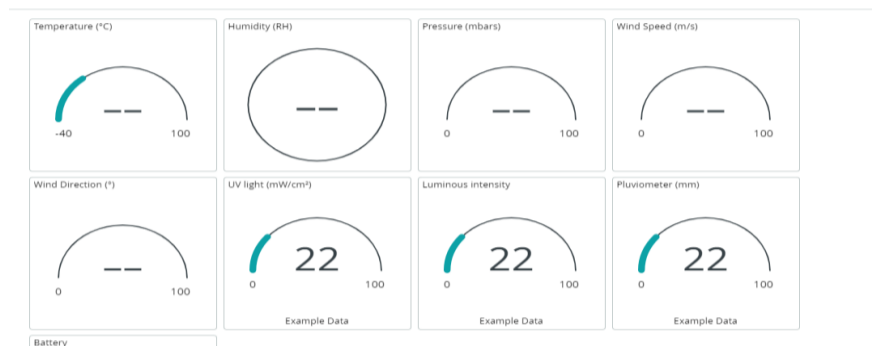


Ilustración 38 Dashboard.

Fuente: Arduino Cloud

8.2 Codificación

A continuación, se muestra parte de la estructura del código implementado para el sistema de riego:

Para esta primera parte, se incluyeron librerías como: “AccelStepper.h” para el control de los motores paso a paso, “Preferences.h” para guardar datos en la memoria Flash del ESP32 y “thingProperties.h”, para asignar cada una de las variables presentes en el Arduino IoT Cloud.

```
#include <AccelStepper.h>
#include <Preferences.h>
#include "thingProperties.h"

#define STEP2 22
#define DIR2 1 //DIR: High = izquierda. Low = derecha
#define ENA2 23 //ENA: High = apagar. Low = prender
#define STEP1 21
#define DIR1 19
#define ENA1 3
#define RIE 16
#define FCY 4 //FC1
#define FCX 15 // FC2
#define BOMB 18 //salida de bomba
#define FLOTA 17 //puerto futuro para flotador (sensor de nivel de agua)
#define LUZ 5 //puerto futuro para luz de cultivo(Luz que nos permite germinar en interior)
#define pwm_motor 0 //Canal de pwm
#define freq_pwm 1000 //frecuencia de pwm motor
#define res_pwm 10 //Resolucion del pwm
```

Ilustración 39 Definición de librerías.

Fuente: Autor.

En la segunda parte del código, se configuran los pines como salidas o entradas, algunos con función de PULL UP y adicional, se denominan los pines para los motores paso a paso, la salida de la bomba, el botón de riego y los dos finales de carrera.

```
pinMode(STEP1, OUTPUT);
pinMode(DIR1, OUTPUT);
pinMode(ENA1, OUTPUT);
pinMode(STEP2, OUTPUT);
pinMode(DIR2, OUTPUT);
pinMode(ENA2, OUTPUT);
pinMode(BOMB, OUTPUT);
pinMode(RIE, INPUT_PULLUP);
pinMode(FLOTA, INPUT_PULLUP);
pinMode(FCY, INPUT_PULLUP);
pinMode(FCX, INPUT_PULLUP);

ledcSetup(pwm_motor, freq_pwm, res_pwm);
ledcAttachPin(BOMB, pwm_motor);
ledcWrite(pwm_motor, 0);
```

Ilustración 40 Definición de pines.

Fuente: Autor.

Nota: La última trama de datos enviada del servidor al ESP 32, es la información que queda guardada en la memoria flash de la misma tarjeta.

Con respecto a la función “guardar_variables”, se realiza la lectura de la memoria flash y cuando se resetea el ESP se hace la lectura de las variables.

```
void guardar_variables() {  
  x_hoyos = memoria.getUChar("xhoyos", 0);  
  y_hoyos = memoria.getUChar("yhoyos", 0);  
  distan = memoria.getUChar("distancia", 0);  
  t_riego = memoria.getUInt("tiemporiego", 0);  
  p_bomba = memoria.getUChar("bomba", 0);  
  n_riego = memoria.getUChar("vecesriego", 0);  
}
```

Ilustración 41 Configuración de la función “guardar_variables”.

Fuente: Autor.

Para la siguiente parte, la función “config_motores” permite la configuración de los motores paso a paso, y se inicia la conexión con el servidor del Arduino Cloud. Dentro de la función, se organiza el Pin de Enable y la configuración del mínimo ancho de pulso que puede enviar, adicional, también se configuran la velocidad máxima y aceleración de ambos motores.

```
void config_motores() {  
  
  motor1.setEnablePin(ENA1);  
  motor1.setPinsInverted(true, false, true);  
  motor1.setMinPulseWidth(20);  
  motor1.disableOutputs();  
  
  motor2.setEnablePin(ENA2);  
  motor2.setPinsInverted(false, false, true);  
  motor2.setMinPulseWidth(20);  
  motor2.disableOutputs();  
  
  motor1.setMaxSpeed(300);  
  motor1.setAcceleration(1000);  
  motor2.setMaxSpeed(300);  
  motor2.setAcceleration(1000);  
  
}
```

Ilustración 42 Configuración de la función “config_motores”

Fuente: Autor.

La función llamada “calibración”, sirve para que, al ser llamada en cualquier apartado del código, realice el ajuste y lleve siempre al mismo valor inicial. Ejemplo: Si el deslizador central en su última labor queda en la mitad del recorrido, al reiniciar el ESP 32, se evalúa dentro del código la función de calibración y de allí el deslizador central se posiciona en las coordenadas iniciales (0,0)

```
void calibracion() {
  motor2.enableOutputs();
  motor1.enableOutputs();
  motor1.setSpeed(-300);
  motor2.setSpeed(300);
  while (digitalRead(FCY)) {
    motor1.runSpeed();
    motor2.runSpeed();
  }
  motor1.stop();
  motor2.stop();
  motor1.setSpeed(-300);
  motor2.setSpeed(-300);
  while (digitalRead(FCX)) {
    motor1.runSpeed();
    motor2.runSpeed();
  }
  motor1.stop();
  motor2.stop();
  motor1.setCurrentPosition(0);
  motor2.setCurrentPosition(0);

  motor1.disableOutputs();
  motor2.disableOutputs();
}
```

Ilustración 43 Configuración de la función “calibración”
Fuente: Autor.

La función “regar”, se activará cuando se oprima el botón de regado, o cuando desde el servidor se oprima el swiche de regado. Además, dentro de este comando, se realiza la conversión de la distancia entre los Hoyos en X y Y a pasos. Ejemplo: Si la distancia entre dos locaciones (hoyos) es de 40mm, equivaldría a 200 pasos del motor.

```

void regar() {
  calibracion();
  int hx = x_hoyos - 1;
  int hy = y_hoyos - 1;
  int steps = distan * 5;
  motor1.enableOutputs();
  motor2.enableOutputs();
  for (int x = 0; x <= hx; x++) {
    xAdelante(steps, x);
    if (x % 2 == 0) {
      motor1.setCurrentPosition(0);
      motor2.setCurrentPosition(0);
      for (int y = 0; y <= hy; y++) {
        yAdelante(steps, y);
        ledcWrite(pwm_motor, p_bomba * 10); //Potencia de la bomba
        delay(t_riego); //Tiempo de riego de la bomba
        ledcWrite(pwm_motor, 0);
        ArduinoCloud.update();
      }
    }
    else {
      motor1.setCurrentPosition(hy * steps);
      motor2.setCurrentPosition(-hy * steps);
      for (int y = hy; y >= 0; y--) {
        yAdelante(steps, y);
        ledcWrite(pwm_motor, p_bomba * 10); //Potencia de la bomba
        delay(t_riego); //Tiempo de riego de la bomba
        ledcWrite(pwm_motor, 0);
        ArduinoCloud.update();
      }
    }
    motor1.setCurrentPosition(steps * x);
    motor2.setCurrentPosition(steps * x);
  }
  ledcWrite(pwm_motor, 0);
  calibracion();
}

```

Ilustración 44 Configuración de la función “regar”

Fuente: Autor.

Además, dentro de esta función para el riego, como se observa en la anterior ilustración, se configuran los hoyos tanto en x como en y, para el desplazamiento, ya sea para el avance o el retroceso. Sin embargo, se debe llevar un control del pwm, ya que por cada posición se debe realizar una parada con un determinado tiempo, que es definido desde el servidor.

```

void yAdelante(int pasitos, int n) {
  motor1.moveTo(pasitos * n);
  motor2.moveTo(-pasitos * n);
  while (motor1.currentPosition() != motor1.targetPosition() && motor2.currentPosition() != motor2.targetPosition()) {
    motor1.run();
    motor2.run();
  }
  motor1.stop();
  motor2.stop();
}

void xAdelante(int pasi, int h) {
  motor1.moveTo(pasi * h);
  motor2.moveTo(pasi * h);
  while (motor1.currentPosition() != motor1.targetPosition() && motor2.currentPosition() != motor2.targetPosition()) {
    motor1.run();
    motor2.run();
  }
  motor1.stop();
  motor2.stop();
}

```

Ilustración 45 Desplazamiento en X y Y.

Fuente: Autor.

Las funciones “yAdelante” y “xAdelante”, vistas en la Ilustración 45, van dirigidas al movimiento en las coordenadas que se determinen, pero están inmersas en la función “regar”, ya que permiten el desplazamiento del deslizador central para realizar la acción del riego.

Si se realiza algún cambio en el servidor, se envía la información al ESP32 y se activa la función específica de cada una de las variables. Por ejemplo, si se cambia la distancia entre hoyos y se pasa de 40mm a 50mm, el servidor lo detecta, envía la información, se evalúa dentro de la función específica y es guardada de inmediato en la memoria flash del ESP32.

```
void onXHoyosChange() {
  // Add your code here to act upon XHoyos change
  memoria.putUChar("xhoyos", x_hoyos);
  memoria.putUChar("distancia", distan);
  memoria.putUChar("yhoyos", y_hoyos);
  memoria.putUInt("tiemporiego", t_riego);
  memoria.putUChar("bomba", p_bomba);
  memoria.putUInt("vecesriego", n_riego);
}

/*
  Since YHoyos is READ_WRITE variable, onYHoyosChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onYHoyosChange() {
  // Add your code here to act upon YHoyos change
  memoria.putUChar("xhoyos", x_hoyos);
  memoria.putUChar("distancia", distan);
  memoria.putUChar("yhoyos", y_hoyos);
  memoria.putUInt("tiemporiego", t_riego);
  memoria.putUChar("bomba", p_bomba);
  memoria.putUInt("vecesriego", n_riego);
}
```

Ilustración 46 Cambios en el servidor.

Fuente: Autor.

Adicionalmente, hay un archivo que es “Thing Properties”, allí están almacenadas las variables creadas en el servidor, su naturaleza y la configuración de la conexión Wifi para que el ESP32 esté conectado en todo momento, como se ve a continuación:


```

#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>

const char DEVICE_LOGIN_NAME[] = "00f38cea-f54c-4426-9fdd-c08bd6eaaf32";

const char SSID[] = SECRET_SSID; // Network SSID (name)
const char PASS[] = SECRET_OPTIONAL_PASS; // Network password (use for WPA, or use as key for WEP)
const char DEVICE_KEY[] = SECRET_DEVICE_KEY; // Secret device password

void onDistanChange();
void onNRiegoChange();
void onPBombaChange();
void onTRiegoChange();
void onXHoyosChange();
void onYHoyosChange();
void onSprayChange();

int distan;
int n_riego;
int p_bomba;
int t_riego;
int x_hoyos;
int y_hoyos;
bool spray;

void initProperties(){
    ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
    ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
    ArduinoCloud.addProperty(distan, READWRITE, ON_CHANGE, onDistanChange);
    ArduinoCloud.addProperty(n_riego, READWRITE, ON_CHANGE, onNRiegoChange);
    ArduinoCloud.addProperty(p_bomba, READWRITE, ON_CHANGE, onPBombaChange);
    ArduinoCloud.addProperty(t_riego, READWRITE, ON_CHANGE, onTRiegoChange);
    ArduinoCloud.addProperty(x_hoyos, READWRITE, ON_CHANGE, onXHoyosChange);
    ArduinoCloud.addProperty(y_hoyos, READWRITE, ON_CHANGE, onYHoyosChange);
    ArduinoCloud.addProperty(spray, READWRITE, ON_CHANGE, onSprayChange, 1);
}

WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);

```

Ilustración 47 Archivo “thingProperties.h”

Fuente: Autor.

RESULTADOS

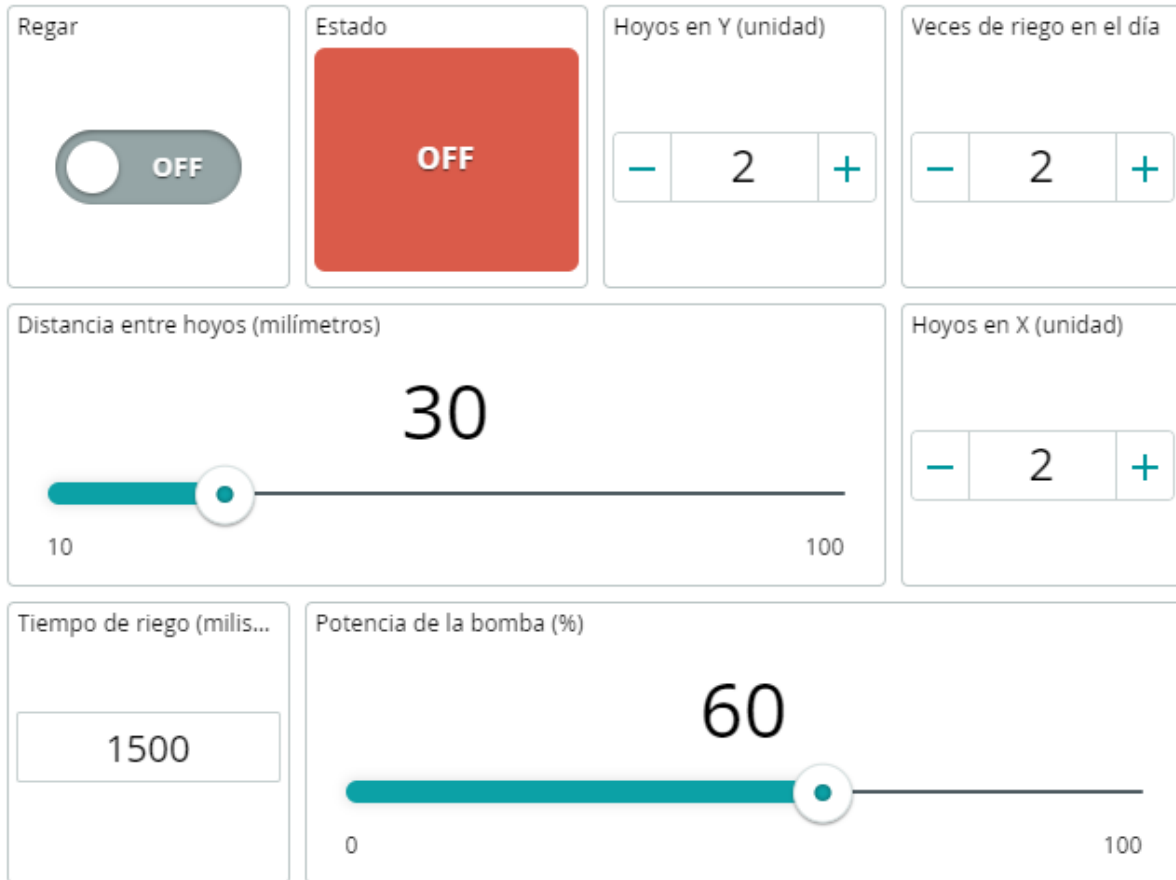


Ilustración 48 Interfaz de usuario funcional.
Fuente: Autor.

La información para el control de sistema que se ve organizada en la Ilustración 48, es la interfaz de usuario final y funcional para el control del sistema de riego por goteo aéreo. En este HMI se controla la potencia de la bomba, el número de veces de riego, la distancia exacta entre los germinadores (hoyos) y el tiempo en milisegundos por cada aspersión realizada.

En la sección central de nombre: "Distancia entre hoyos", se establece a que distancia está un compartimiento entre otro, esto permite calcular la malla y realizar un riego preciso en cada uno, además, junto con las secciones que se ven en la esquina superior derecha: "Hoyos en Y" y "Hoyos en X", se configura la máquina para agregar el número

de compartimientos que se tiene a disposición, ya sea una bandeja personalizada o una estándar, como se puede apreciar en la siguiente imagen:

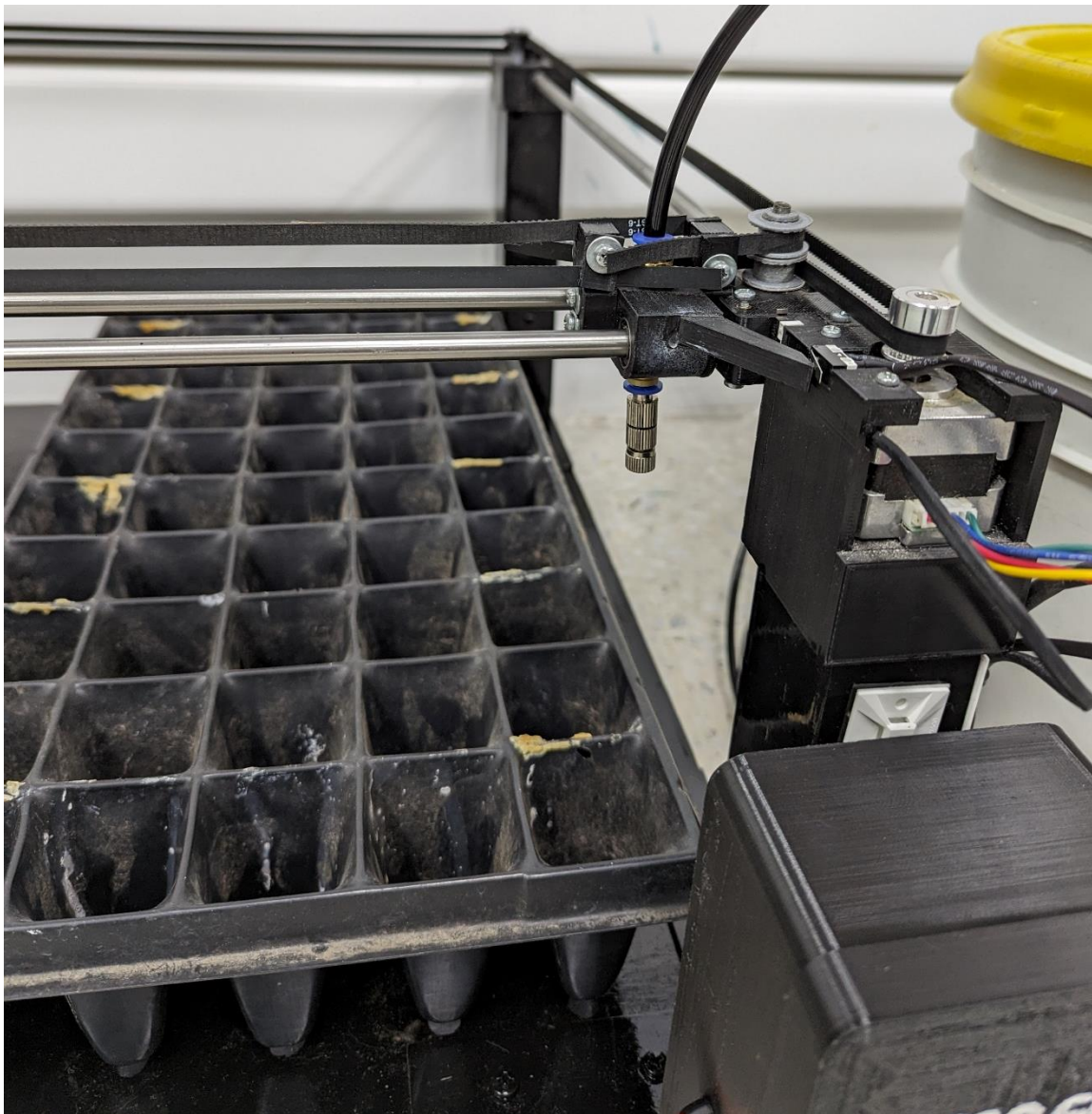


Ilustración 49 Bandeja de germinadores utilizada para el proyecto.

Fuente: Autor.

Con respecto al diseño planteado para el mecanismo, se cumplieron con los objetivos: los motores paso a paso permiten el movimiento del deslizador central que es el que lleva la boquilla para el riego, cumpliendo con el principio de funcionamiento CORE XY, como se evidencia en la siguiente imagen:

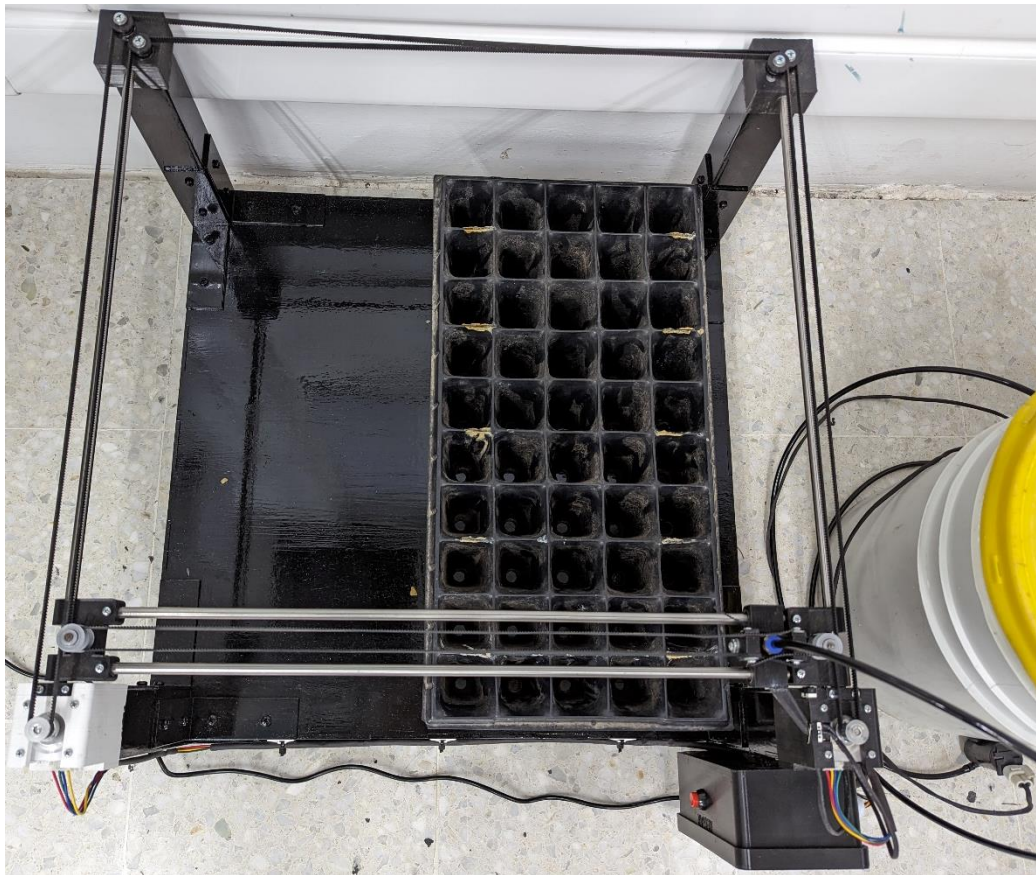


Ilustración 50 Mecanismo de riego Core XY.

Fuente: Autor.

Cabe destacar, que el tiempo de riego se determina a través del HMI, el deslizador central debe detenerse cierta cantidad de tiempo en cada compartimiento para realizar el riego, además del tiempo, también se debe configurar el nivel de potencia, esto evita que el agua a través de la boquilla salga a grandes chorros y dañe la semilla, y permite tener un mayor control sobre la bomba.

Cómo se puede observar en la siguiente Ilustración 51, se dispone de una caneca que hace la función de almacenamiento de agua. En su parte inferior, va acoplada la bomba junto con una goma de protección (antiderrame), que como su nombre lo indica, evita que se generen pérdidas del líquido. De allí, el agua pasa a través de una manguera de 6mm al deslizador central.



Ilustración 51 Almacenamiento de agua con acople de la bomba.

Fuente: Autor.

Y se puede evidenciar, en la siguiente imagen, la construcción de una caja para la inclusión del circuito electrónico del sistema de riego. Sin embargo, se debe considerar lo siguiente:

Si en algún momento, se presenta alguna falla con el HMI, el usuario tendrá control manual de motor de regado, pero se cumplirá con la última operación establecida y guardada en la tarjeta de adquisición, en este caso: el ESP 32.

Ejemplo: Si en la última operación realizada, se establecieron 15 hoyos, 5 en X y 10 en Y, más la malla de trabajo, esa será la operación por repetir en caso tal de tener el inconveniente con la interfaz.

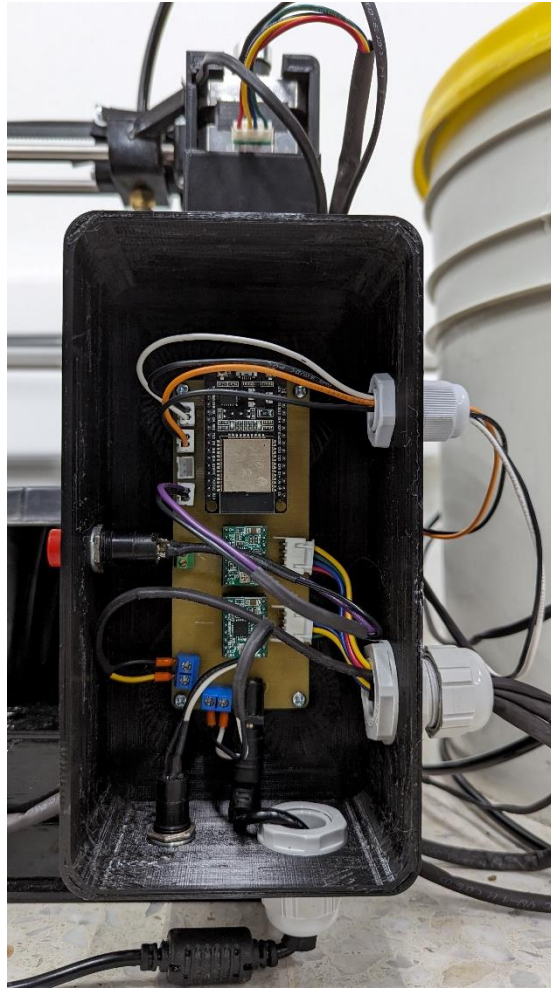


Ilustración 52 Circuito electrónico sistema de riego por goteo aéreo.
Fuente: Autor.

DISCUSIÓN

En un principio se habló de aspersión por nebulización, pero al realizar el ensamble y poner el sistema en funcionamiento, ocurrió lo inesperado: el agua se pulverizaba bastante y, por tanto, como parte del fluido se esparcía en el aire, se desperdiciaba bastante, y fuera de ello, requería un consumo mayor de energía. Sin embargo, parte de la solución era utilizar una alimentación mejor para el sistema.

Analizando la situación presentada y los múltiples inconvenientes, se optó por utilizar una bomba más pequeña, como se puede ver en el apartado anterior de Resultados, con menor presión para poder hacer el riego a través de una boquilla y así evitar pérdidas, ahorrar en consumo de energía y generar un sistema más eficiente.

HERRAMIENTAS Y COSTOS

Artículo	Precio (COP)
ESP32	\$ 35.000
Driver Motor PAP (A4988) X2	\$ 30.000
Piezas impresas en 3D PLA	\$ 150.000
Transistor FDD8580 SMD D-pack	\$ 15.000
Bomba de agua 12V Poong Sung 2I14	\$ 70.000
Regulador 5v LDO LD1117 SMD	\$ 5.000
Resistencias	\$ 5.000
Conector JST XH2.54 Hembra (4 pin)	\$ 5.000
Conector JST PH2.0 Hembra (2 pin)	\$ 3.000
Conector JST XH2.54 Macho (4 pin)	\$ 5.000
Conector JST PH2.0 Macho (2 pin)	\$ 3.000
Motor paso a paso 17HS3401 x2	\$ 120.000
Refuerzos para la base	\$ 60.000
Conector acople rápido PC6-01 x2	\$ 20.000
Correa GT2 5 metros	\$ 30.000
Poleas GT2 20 dientes x2	\$ 20.000
PCB virgen 10cmx15cm	\$ 10.000
Finales de carrera	\$ 4.000
Tornillería	\$ 25.000
Fuente 12v 5a	\$ 60.000
Diodo Schottky SS34 SMD	\$ 10.000
LED 1205 SMD	\$ 1.000
Capacitores tantalio SMD	\$ 5.000
Bornera 4mm x2	\$ 6.000

Artículo	Precio (COP)
Plug Jack 2.5mm	\$ 2.500
Pulsador switch rojo	\$ 3.000
Pulsador rojo	\$ 3.000
Cableado 22 awg	\$ 20.000
Plug Jack 2.5mm	\$ 2.500
Suscripción Arduino IoT Cloud	\$30.649
TOTAL	\$ 747.500

Tabla 1 Herramientas y costos

CONCLUSIONES

1. El diseño de un sistema de riego puede resultar complejo si dentro de los parámetros, no priman la presión de la bomba y de la boquilla, ya que si hay mucha presión en la aspersión se pueden generar huecos en el área de las semillas y puede afectar el proceso de germinación.
2. Hacer la estructura del sistema de riego con perfiles de aluminio, permite dar mayor soporte y mejorar el mecanismo Core XY. El aluminio es un material ligero y resistente, esto reduce el peso total del mecanismo, lo cual puede ser beneficioso en términos de eficiencia energética y capacidad de carga, además cuando se usan los perfiles, se garantiza una alineación y posicionamiento precisos de los componentes del mecanismo CORE XY. Esto es crucial para lograr un movimiento suave y preciso del deslizador central del sistema de riego.
3. Este sistema de riego tiene la gran ventaja de ser controlado de forma remota, a través del servidor de Arduino Cloud y desde cualquier parte en la que se tenga conexión a internet. Por otro lado, el código puede ser manipulado y cargado a la tarjeta de adquisición, en este caso el ESP32, sin necesidad de estar junto a la máquina. Esta actualización del firmware se hace a través del servidor, aunque debe verificarse que la instrucción dada sea realizada de la manera correcta en el sistema. Finalmente, siempre habrá que tener conexión a internet para generar cambios dentro de la estructura del código, para interactuar con la máquina o para ser operada mientras se realice algún cambio, adicionalmente y como factor primordial, también deberá estar conectado a la energía para que pueda realizarse su óptima operación.
4. Teniendo en cuenta que, en otros sistemas de riego convencionales, hay muy poco desarrollo en el área de germinadores o por lo menos, de sistemas más

robustos para cultivos específicos, el sistema de riego autónomo desarrollado tiene justo la ventaja de que, dependiendo de las instrucciones dadas a través del código, la maquina desarrolla su trabajo, evitando que un operario deba hacer todo el proceso de forma manual. Este avance genera mayor precisión en el riego para cada germinador, la cantidad de agua exacta o requerida, mayor cuidado de la semilla y, además, reducción en costos y tiempo de operación.

5. En otros sistemas convencionales, utilizan una numerosa cantidad de mangueras y llaves por cada germinador, lo que genera mayores costos y contaminación, debido a que llevan mayor cantidad de plásticos, que no son reutilizables y que demoran una eternidad en descomposición. En cambio, en este sistema de riego, se utilizan materiales como el metal para los soportes y demás piezas son impresas en PLA, un material amigable con el medio ambiente, biodegradable y, además, reutilizable. Se evita el uso de cantidad de mangueras, ya que sólo hay una con boquilla que se desplaza a través del deslizador central y genera el riego a cada germinador.

RECOMENDACIONES

1. Mejorar el diseño de la PCB para generar un diseño más ergonómico y organizado.
2. Implementar sensores de humedad, presión y demás que permitan tener un mayor control del sistema y se tenga mayor precisión en el control de riego.
3. Diseñar y/o adaptar una boquilla que permita un riego más suave y no le genere algún daño a la semilla.
4. Incluir un flotador o un sensor electrónico de nivel, para el depósito de agua, con el fin de generar una alarma por bajo nivel del recurso hídrico.
5. El mantenimiento debe ser por lo mínimo cada 15 días, para realizar la limpieza de los rodamientos, evitando la acumulación de polvo que podría generar un mayor desgaste de las varillas por donde hacen su recorrido. Se deben también lubricar las piezas que lo requieran, verificar el funcionamiento de la bomba para que nada le genere obstrucción, revisar la boquilla de aspersion y verificar el nivel del tanque de agua.
6. Colocar un filtro a la bomba para que no pasen contaminantes por la misma.

REFERENCIAS

- [1] I. A. R. A. y. M. E. Miserendino, *Invernaderos: Aspectos básicos sobre estructura, construcción y condiciones ambientales*, Agricultura 23, 2014.
- [2] NOVAGRIC, «Novedades Agrícolas S.A.», 2022. [En línea]. Available: <https://novagric.com/es/venta-invernaderos-novedades/tipos-de-invernaderos/invernadero-tunel>.
- [3] P. P. y. T. Granollers, «Diseño y Automatización Industrial,» Universidad Politécnica de Cataluña, 2013.
- [4] W. R. Lozada, «Diseño y construcción de un Germinador,» Universidad Tecnológica de Pereira, Pereira, 2016.
- [5] C. G. S. Ruíz, «Germinación de semillas: Manual de divulgación para uso en instituciones de educación.,» Toluca, Estado de México.
- [6] J. C. A. Tobón, «Métodos de riego.,» Universidad Nacional de Colombia, Medellín.
- [7] L. A. Gurovich, *Fundamentos y diseño de sistemas de riego.*, San José, Costa rica: IICA, 1985.
- [8] L. J. C. T. D. K. P.-M. Alfonso A. GUIJARRO-Rodríguez, «Sistemas de riego automatizado con Arduino,» *Espacios*, vol. 39, nº 37, p. 27, 2022.
- [9] I. & S. B. García Casillas, «Sistemas de riego por aspersión y goteo,» 2015.
- [10] Grupo Hidráulica, «Los mejores sistemas automatizados para el riego,» 20 Octubre 2021. [En línea]. Available: <https://n9.cl/eb17m>.
- [11] A. S. Valencia, «Diaporama Sistemas de riego por nebulización,» Universidad Autónoma del Estado de México, México, 2018.
- [12] J. P. G. N. O. L. G. M. ,. L. C. Tovar Soto, «Propuesta de diseño de una red inalámbrica de sensores y actuadores para riego, con tecnología de Internet de las Cosas,» *investigación e innovación en ingenierías.* , vol. 10, nº 1, 30 Abril 2022.
- [13] J. Monserrat, «Sistemas de Riegto para uso en viveros.,» *Horticom*, 2005.
- [14] H. J. M. Angeles, «Algunos Aspectos de Diseño, Operación, Mantenimiento y Evaluación de los Sistemas de Riego por Goteo,» *INTAGRI*, nº 31, 2019.
- [15] NETAFIM, «El riego por goteo revoluciona la agricultura.,» 2023. [En línea]. Available: <https://www.netafim.co/riego-por-goteo/>.
- [16] Amazon, «uxcell Pneumatic Air Tubing, 6mm OD x 4mm ID 6m(236 Inch) PU Polyurethane Air Compressor Tubing Hose Pipe Blue,» 2023. [En línea]. Available: <https://n9.cl/1710c>.

- [17] S. T. F. V. S. Flavio Capraro, «Laboratorio Virtual y Remoto para Simular, Monitorizar y Controlar un Sistema de Riego por Goteo en Olivos,» *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 7, nº 1, 2010.
- [18] I. E. Moyer, «CORE XY,» 2012. [En línea]. Available: <https://corexy.com/index.html>.
- [19] V. A. Armijos, «Electronics,» Módulo Adafruit ESP32, 2023. [En línea]. Available: <https://vasanza.blogspot.com/2021/07/especificaciones-del-modulo-esp32.html>.
- [20] Movil Electronics, «Motor NEMA 17 17HS3401 1.3A,» 2023. [En línea]. Available: <https://moviltronics.com/tienda/pap-17hs3401/>.
- [21] FireChild Semiconductor, «Datasheet Transistor FDD8580 SMD D-pack,» 2006.
- [22] STMicroelectronics, «LD1117 Datasheet».
- [23] Arduino IoT Cloud, «Plataforma,» [En línea]. Available: <https://docs.arduino.cc/arduino-cloud/>.

ANEXOS

Anexo 1: Guía para el uso de la plataforma de *Arduino Cloud*

Paso #1: Se debe ingresar al siguiente Link: <https://cloud.arduino.cc/> y dar clic en la opción: “Get Started” como se observa en la siguiente imagen:

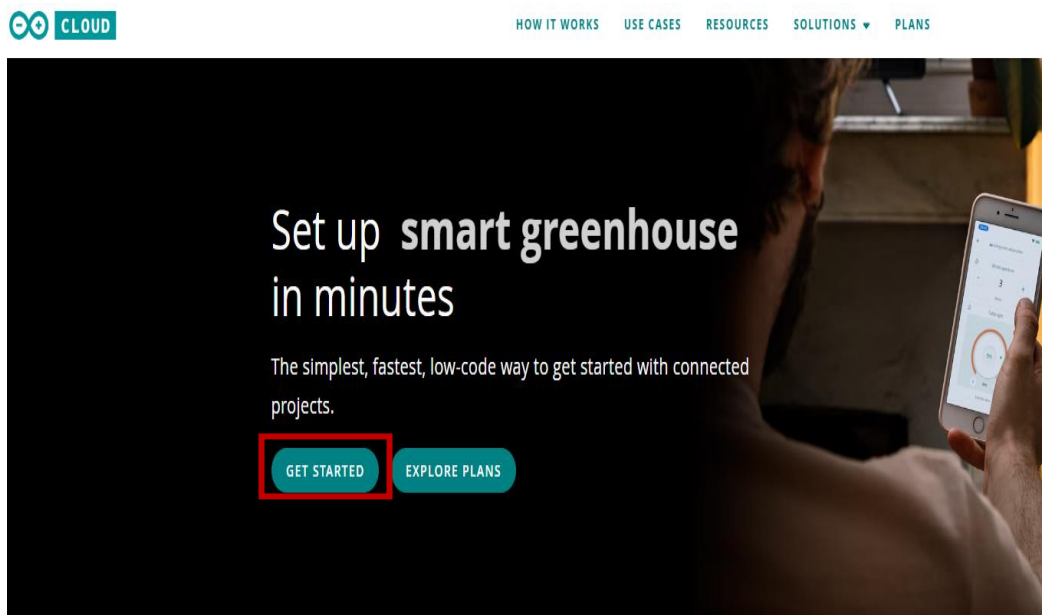


Ilustración 53 Inicio plataforma Arduino Cloud.

Fuente: Arduino Cloud.

Después se habilitará una nueva pestaña de acceso, como a continuación se muestra y se deberá presionar en el botón “Sign In”; si no hay una cuenta, puede crear una y allí le pedirán datos como:

- Fecha de nacimiento.
- Correo electrónico.
- Contraseña (y confirmación).
- Y aceptar los términos y condiciones.

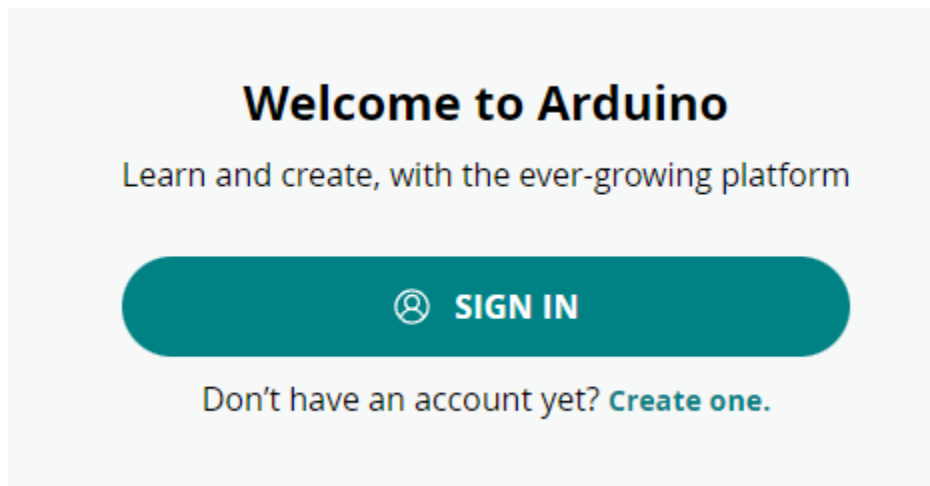


Ilustración 54 Pestaña de acceso Arduino Cloud.

Fuente: Arduino Cloud.

Si la cuenta ya está creada, bastará con colocar correo electrónico y contraseña o hacer una conexión directa con la cuenta de Google, GitHub, o la cuenta de uso frecuente en redes sociales como Facebook. Ver la siguiente ilustración:

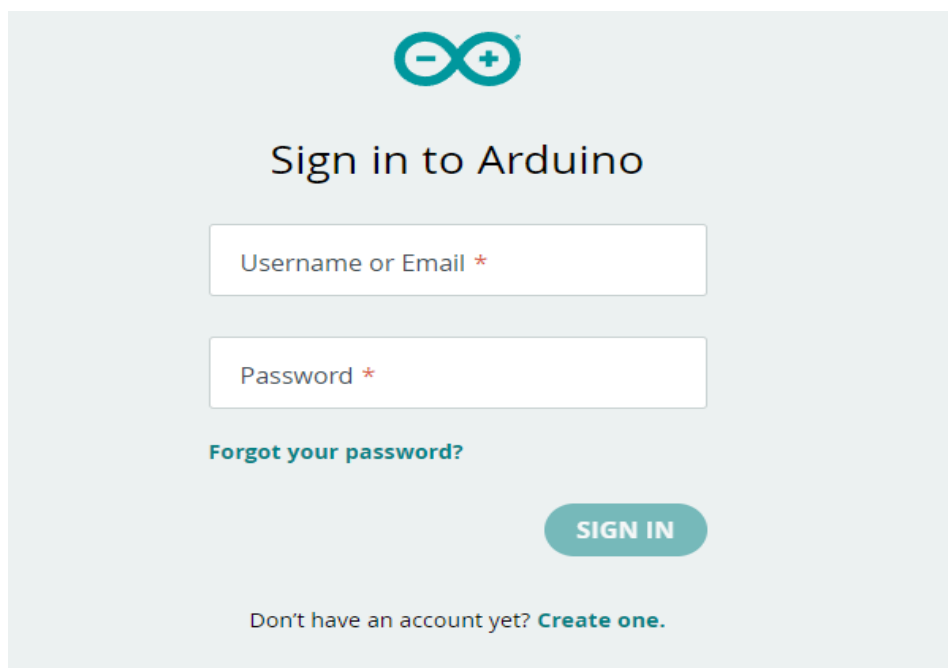


Ilustración 55 Cuenta de acceso Arduino Cloud.

Fuente: Arduino Cloud.

NOTA: Se dejará al operario o persona a cargo del sistema las credenciales para ingreso a la Interfaz creada, tanto correo como contraseña, esto con el fin de que sólo la persona autorizada tenga control del sistema y se pueda tener seguridad dentro del mismo.

Al acceder a la plataforma, se abrirá la ventana que se ve a continuación y en ella aparecerán las opciones de lo que se desea realizar, ya sea codificar, o crear una nueva área de trabajo u obtener documentación sobre Arduino IoT Cloud.

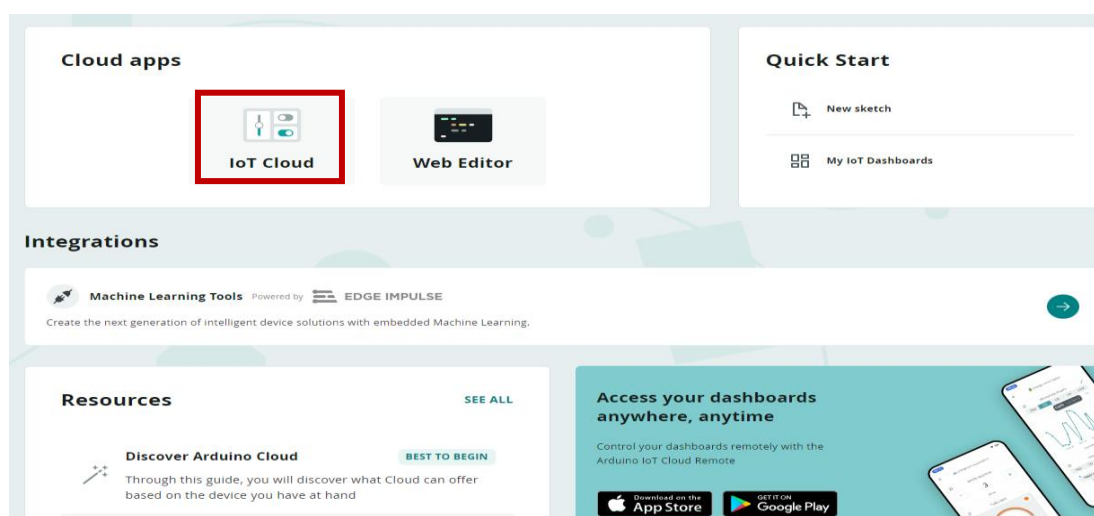


Ilustración 56 Menú inicial Arduino Cloud.

Fuente: Arduino Cloud.

Luego, se dará clic sobre IoT Cloud como se observa en la ilustración anterior y esto, permitirá abrir los proyectos ya creados o los nuevos.

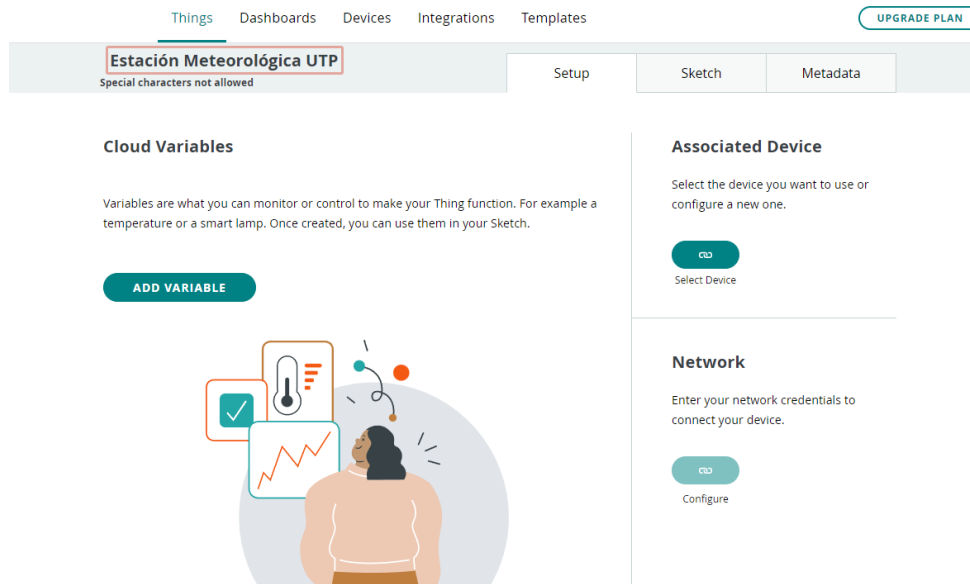


Ilustración 57 Ventana de configuración de dispositivos.

Fuente: Arduino Cloud.

Después de seleccionar el botón de IoT Cloud, se desplegará una ventana con todas las herramientas para trabajar en la plataforma, en la sección superior se puede encontrar la opción “Things”, que es para configurar nuevos dispositivos, como sensores, y sus respectivas variables, en seguida se puede observar “Dashboard”, el cual será el área de trabajo para crear el HMI, continuo está “Devices”, donde se configuran las nuevas placas de conexión, ya sean de Arduino u otro Hardware compatible y finalmente, “Integrations” y “Templates” donde se puede dar un efecto artístico, más amigable con el usuario y con múltiples opciones de trabajo.

A continuación, se muestran algunas ilustraciones como ejemplo de lo que grosso modo puede realizarse en la plataforma:

Things Dashboards Devices Integrations Templates

Sensor Temperatura Setup

Cloud Variables **ADD**

Name ↓	Last Value	Last Update
<input type="checkbox"/> Temperatura <small>bool temperatura;</small>	-	

Ilustración 58 Configuración de sensor y especificación de variable.
Fuente: Arduino Cloud.

Add variable

Name
Temperatura

Sync with other Things i

Boolean eg, true

Declaration i
`bool temperatura ;`

Variable Permission i

Read & Write Read Only

Variable Update Policy i

On change Periodically

ADD VARIABLE CANCEL

Ilustración 59 Configuración de las variables
Fuente: Arduino Cloud.

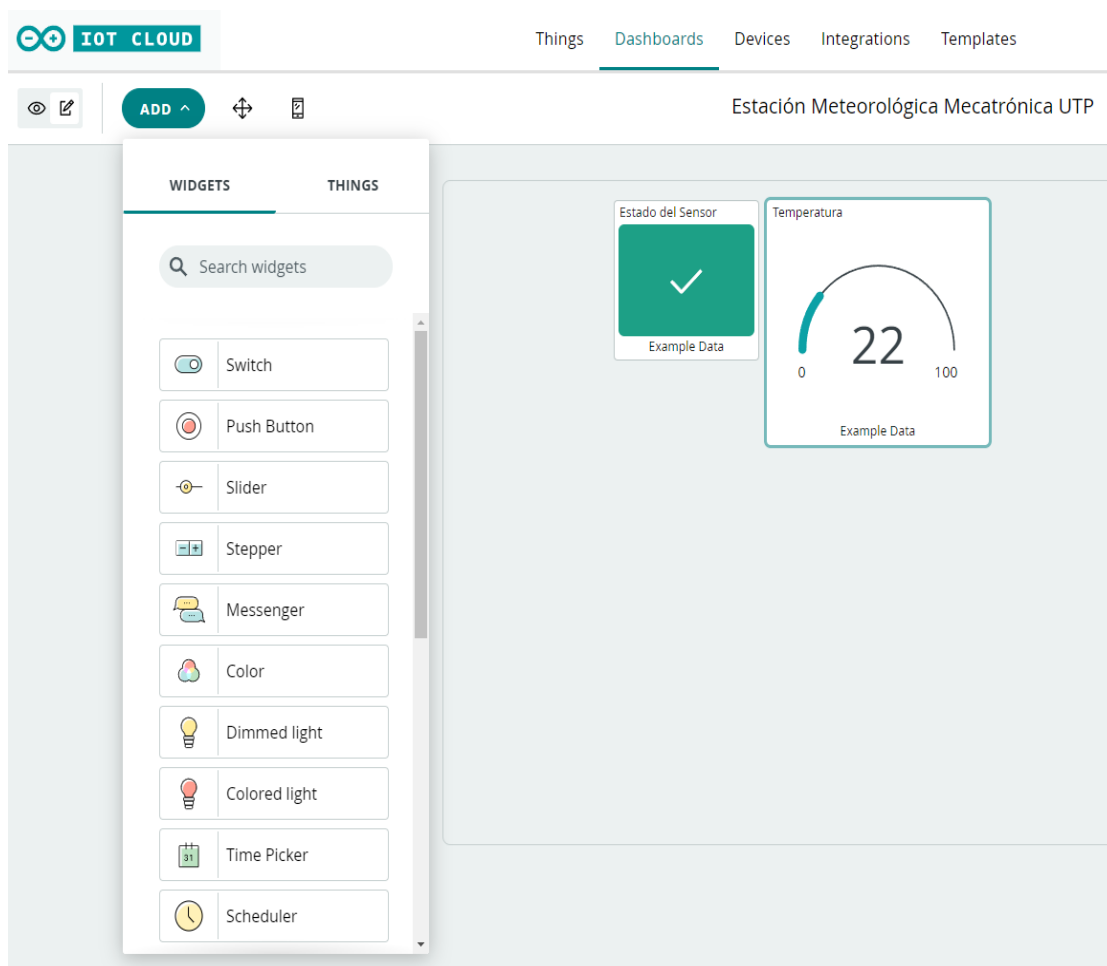


Ilustración 60 Ejemplo de diseño de Dashboard.

Fuente: Arduino Cloud.