

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

SISTEMA DE RIEGO PARA INVERNADEROS CON INTERFAZ WEB UTILIZANDO UN ARDUINO YUM

SERVIDOR WEB EN ARDUINO YUM

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

RICARDO XAVIER CHASI RAMOS

ricardo.chasi@epn.edu.ec

DIRECTOR: RAMIRO EDUARDO MOREJÓN TOBAR

ramiro.morejon@epn.edu.ec

DMQ, octubre 2023

CERTIFICACIONES

Yo, Ricardo Xavier Chasi Ramos declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Ricardo Xavier Chasi Ramos

Certifico que el presente trabajo de integración curricular fue desarrollado por Ricardo Xavier Chasi Ramos, bajo mi supervisión.

Ramiro Eduardo Morejón Tobar
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

RICARDO XAVIER CHASI RAMOS

RAMIRO EDUARDO MOREJON TOBAR

DEDICATORIA

A mis padres que a pesar de todas las adversidades que hemos afrontado en el transcurso de la vida, me han apoyado en todo lo que este en su alcance para culminar este gran primer logro.

A mis hermanos y hermanas que me han cuidado, guiado y me han dado todas las oportunidades del mundo para poder enfrentar el gran camino que fue la universidad y darme las ganas para nunca rendirme... les quedaré debiendo esta vida y la otra.

A la persona que siempre ha estado desde que entre a la universidad y me ha proyectado a ser mejor persona, te amo con mi vida.

AGRADECIMIENTO

Agradezco de todo corazón a Dios y a la vida por permitirme tener una madre tan buena y amorosa que sigue estando conmigo en las buenas y en las malas, la que estará más orgullosa por obtener el título que yo mismo, tal como fue en el jardín, escuela y colegio. Madrecita de mi corazón gracias por todo, te amo como a nadie más en este mundo.

A mi padre, que siempre me ha guiado por el camino del bien para ser un mejor profesional, hijo, hermano y persona.

A mis hermanos y hermanas, en especial a Diana y Lorena, que fueron mis segundas madres durante el transcurso de la universidad, sin ellas no hubiera logrado lo que tanto anhelaban de mí, ser un gran profesional.

A mi enamorada, Salomé, que estuvo conmigo desde la primera instancia que pise la universidad hasta la actualidad, sin ella me hubiera vuelto loco en este trayecto como estudiante y como profesional.

A mis amigos de la universidad Carlos, Cristian, Francisco, Alejandra y todos lo que lleguen a leer este párrafo, gracias de todo corazón por hacer de la vida universitaria más amena y feliz a pesar de los altos y bajos que hemos tenido en nuestras carreras. Todas las experiencias y momentos que hemos pasado nunca las olvidare. Gracias muchachos...

A toda mi familia y amigos que he hecho durante mi vida universitaria como profesional y que no logré mencionar, gracias por darme ánimos, gracias por mostrarme las distintas formas de ver la vida, de cómo actuar y de cómo ser. Fueron varias piezas fundamentales en este rompecabezas llamada vida.

Por último, quiero agradecer a la grandiosa Escuela Politécnica Nacional que me ha permitido ser un gran profesional y conocer a los ingenieros que me han acompañado en la etapa final de la carrera al Ing. Ramiro Morejón y al Ing. Diego Reinoso, quienes me han apoyado y me han guiado de la manera más correcta para la culminación de este proyecto.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 INTRODUCCIÓN	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	3
1.4.1 INVERNADERO	3
1.4.2 VARIABLES AMBIENTALES PARA MONITOREAR DENTRO DE UN INVERNADERO	3
1.4.2.1 Temperatura	4
1.4.2.2 Humedad	4
1.4.2.3 Humedad relativa.....	4
1.4.2.4 Nivel de CO2.....	4
1.4.2.5 Luminosidad.....	5
1.4.3 ARDUINO.....	5
1.4.3.1 Arduino UNO.....	5
1.4.3.2 Shield Arduino.....	6
1.4.3.3 Dragino YUM Shield.....	6
1.4.3.4 Arquitectura Dragino YUM Shield.....	7
1.4.4 SERVIDOR WEB Y LINUX.....	8
1.4.5 HTML, CSS Y JAVASCRIPT	9
1.4.5.1 HTML (Hypertext Markup Lenguaje)	9
1.4.5.2 CSS (Cascading Style Sheets).....	9
1.4.5.3 JavaScript.....	9
2 METODOLOGÍA.....	9

2.1	Configuración del shield Dragino Yum	10
2.1.1	ACCESO VÍA SSH	12
2.2	Creación de los archivos de la para la página web	13
2.2.1	Directorio /www/	13
2.2.2	HTML	14
2.2.3	CSS.....	18
2.2.4	JavaScript.....	19
2.3	Base de datos	21
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	24
3.1	Resultados	24
3.1.1	Visualización del servidor web.....	24
3.2	Conclusiones.....	30
3.3	Recomendaciones.....	31
4	REFERENCIAS BIBLIOGRAFICAS	33
5	ANEXOS.....	37
	ANEXO I. Código fuente del archivo index.html.....	37
	ANEXO II. Código fuente del archivo datos.html.	38
	ANEXO III. Código fuente del archivo temperatura.html	41
	ANEXO IV. Código fuente del archivo humedad.html	42
	ANEXO V. Código fuente del archivo humedadrel.html	43
	ANEXO VI. Codigo fuente del archivo nivelco2.html.....	44
	ANEXO VII. Código fuente del archivo luminosidad.html.....	46
	ANEXO VIII. Código fuente del archivo historico.html.....	47
	ANEXO IX. Código fuente del archivo styless.css	50
	ANEXO X. Código fuente del archivo temperatura_grafica.js	56
	ANEXO XI. Código fuente del archivo humedad_grafica.js	57
	ANEXO XII. Código fuente del archivo grafica_humedadrel.js	59
	ANEXO XIII. Código fuente del archivo grafica_co2.js.....	60
	ANEXO XIV. Código fuente del archivo grafica_luminosidad.js.....	61
	ANEXO XV. Código fuente del archivo script.js	63

RESUMEN

En el presente trabajo de integración curricular se contempla el componente para representar gráficamente mediante una página web un sistema de riego para invernaderos utilizando la plataforma Arduino YUM. Específicamente, en este componente se concentra en la visualización de las variables de entornos que influyen en un invernadero, tales como: temperatura, humedad, humedad relativa, nivel de CO₂, intensidad de luz, entre otras; dichas variables pueden ser visualizadas a través de una interfaz web. Para lograr tal propósito se hizo uso de una placa Dragino YUM shield, que tiene un procesador con sistema Linux, con la cual se puede trabajar en conjunto con plataformas de Arduino UNO, DUE o Leonardo, para el presente trabajo se hizo uso del Dragino YUM Shield con el Arduino UNO.

La estructura del presente documento está dividida de tal forma que primero se habla acerca de las características de un invernadero, conceptos más relevantes acerca de la plataforma Arduino Yum y su conexión con el sistema operativo Linux que permite la creación de páginas web.

Seguidamente, se describe la configuración y el código de programación efectuado sobre la plataforma Arduino YUM para lograr visualizar en una página web las variables de entorno del invernadero que fueron mencionadas anteriormente.

Además, se presentan las respectivas evidencias del correcto funcionamiento del componente descrito. Por último, se encuentran las conclusiones y recomendaciones más relevantes encontradas durante la ejecución del presente documento.

PALABRAS CLAVE: Arduino YUM, Invernadero, Linux, Página web.

ABSTRACT

The current project includes a component to graphically represent an irrigation system for greenhouses using the Arduino YUM platform. Specifically, this component focuses on visualizing the environmental variables that influence a greenhouse, such as temperature, humidity, CO₂, and light intensity, which can be displayed through a web interface. To achieve this purpose, Dragino YUM Shield was used, which contains an Arduino Linux processor. It can work together with Arduino UNO, DUE, or Leonardo platforms. For this work, the Dragino YUM Shield was used with the Arduino UNO.

The structure of this document is divided so that it first discusses the characteristics of a greenhouse and the most relevant concepts about the Arduino Yum platform and how is the connection with the Linux operating system, which enables the creation of web pages.

Next, the configuration and programming code performed on the Arduino YUM platform to visualize the greenhouse's environmental variables on a web page are described.

In addition, the corresponding evidence of the proper functioning of the described component is presented. Finally, the most relevant conclusions and recommendations found during the execution of this document are provided.

KEYWORDS: Arduino YUM, greenhouses, Linux, Web Page.

1 INTRODUCCIÓN

Con la invención de los sistemas embebidos a lo largo de la época como son las plataformas Arduino o Raspberry Pi, han permitido desarrollar diversas aplicaciones en diferentes sectores de la sociedad, un claro ejemplo es en el sector agrícola debido a que permiten lograr un ambiente controlado para ciertos cultivos o plantaciones que están adaptados a temperaturas y características específicas para su crecimiento.

En la actualidad, los invernaderos inteligentes han tenido una buena acogida y con esto un gran aumento en su construcción y diseño, debido a que ofrecen varias ventajas, permiten crear un microclima para distintos cultivos, plantas o frutas en un entorno con condiciones controladas como: temperatura, humedad, intensidad de luz y sistema de riego, logrando obtener dichos cultivos fuera de época y en óptimas condiciones [1]. Además, por el diseño de infraestructura semitransparente que suele tener un invernadero se corre un menor riesgo de que los cultivos sufran daños ocasionados por el medioambiente, permiten que la intensidad de luz sea la adecuada para realizar los procesos biológicos de los cultivos, como es la fotosíntesis, permitiendo más productividad y que las condiciones del cultivo al año sean más rentables [2].

Para llevar a cabo el control y el monitoreo de todas las condiciones de un invernadero que se mencionaron anteriormente, se necesita de un dispositivo capaz, que una vez configurado sea posible monitorear y visualizar todas las tareas que conlleva un invernadero inteligente. Dentro del mercado existen varias opciones y una de estas es la plataforma Arduino, un sistema embebido, el cual tiene distintas variantes con características intrínsecas propias de cada uno y por lo general llevan integrado un microcontrolador muy potente para el monitoreo de las variables de entorno, domótica y otros trabajos afines.

Para el documento en cuestión se hace uso de la variante denominada Arduino YUM, para ser específico el Shield Dragino YUM. La característica más importante de esta plataforma es que hace uso de un microprocesador Atheros AR9331 que permite la conexión con el sistema operativo Linux posibilitando la creación de un servidor web para el seguimiento, monitoreo y la presentación de las variables en tiempo real que se tenga en el invernadero. El Shield Dragino YUM es compatible y puede trabajar en conjunto con otras plataformas de Arduino, para el presente trabajo se hizo uso del Shield Dragino YUM con el Arduino UNO permitiendo conectarse a una red Wi-Fi o cableada gracias al panel web que tiene integrado el Arduino YUM [3].

Las variables de entorno que son mostradas y usadas para ser monitoreadas a través de un servidor web configurado y diseñado dentro de la plataforma de Arduino YUM son las siguientes: temperatura, intensidad de luz o luminosidad, humedad, niveles de CO2, y humedad relativa. A lo largo de este trabajo, se detalla a que hace referencia cada una de las variables, la afectación que tiene sobre el invernadero, como son presentadas mediante la interfaz web y su respectivo algoritmo o código.

1.1 Objetivo general

Familiarizarse acerca de las características y funcionamiento de la plataforma Arduino YUM, con el fin de definir una estructura de variables que contenga los parámetros de operación del invernadero, para diseñar una página web utilizando lenguajes HTML, CSS y JavaScript, con el propósito de presentar las variables que influyen en el invernadero.

1.2 Objetivos específicos

1. Conocer acerca de las características y funcionamiento de la plataforma Arduino Yum.
2. Definir una estructura de variables que contengan los parámetros de operación del invernadero.
3. Conocer acerca de las funcionalidades que ofrece la distribución de sistema operativo Linux que ofrece la placa Arduino Yum Shield.
4. Diseñar una página web en base a la estructura de datos y levantar dicho servicio en el Arduino Yum.
5. Codificar la página web diseñada utilizando lenguajes HTML, CSS y JavaScript para subir a la plataforma de Arduino Yum, permitiendo observar las variables del invernadero.
6. Visualizar las variables de entorno del invernadero a través de un servidor web.

1.3 Alcance

Para el presente trabajo se contempla la visualización de las variables de entorno predefinidas para un invernadero a través del diseño de una página web. El sistema embebido que se usa para el siguiente diseño es el Dragino Yum Shield que, comparado con sus otras variantes, tiene integrado un procesador Atheros AR9331, el cual, permite la creación y diseño de una página web. El procesador mencionado anteriormente, tiene una distribución de Linux basada en OpenWrt, con lo cual puede ser configurada mediante una

red Wi-Fi inalámbrica o por medio de un cable Ethernet directo y así lograr acceder a las líneas de comando vía SSH [3] creando los archivos pertinentes para el diseño de la página web en un editor de código para facilitar el diseño.

Para capturar las variables y poder monitorearlas, se crearon varios archivos que deben de ser subidos en el directorio /www/ en la interfaz Linux de la placa con distintos formatos de lenguaje: HTML, CSS y JavaScript, además del uso del IDE de Arduino para verificar la interconexión de las placas. Para comprobar que exista la interconexión anteriormente mencionada entre ambas plataformas o placas usadas en el presente documento, que es un Arduino UNO en conjunto con el Dragino Yum shield se hizo uso de la librería "bridge".

1.4 Marco Teórico

1.4.1 INVERNADERO

Un invernadero se puede definir como una infraestructura cerrada, diseñado con materiales como el plástico o el vidrio que permiten el paso de luz natural; posibilitando crear un ambiente favorable o un entorno controlado a los cultivos que también es denominado como microclima [2]. Entre los principales beneficios que ofrece un invernadero es poder controlar condiciones ambientales y atmosféricas para crear condiciones ideales del cultivo, como la temperatura, humedad, niveles de CO₂, luminosidad entre otras, permitiendo cosechar en temporadas que no son posibles normalmente.

Adicionalmente, un invernadero permite proteger a los cultivos de condiciones climáticas adversas, como son las bajas o altas temperaturas, fuertes lluvias, vientos impetuosos, cantidad de luz, logrando así que un invernadero controlado e inteligente sea utilizado en varios ámbitos laborales como plantas ornamentales, jardinería, sector de alimentos [4].

1.4.2 Variables ambientales para monitorear dentro de un invernadero

Existen una gran cantidad de variables que se pueden monitorear y automatizar dentro de un invernadero, cada una de ellas presenta características que influyen directamente sobre el crecimiento, bienestar y cuidado de los cultivos.

A continuación, se detalla brevemente las variables que son usadas para ser presentadas y visualizadas dentro de la página web; las mismas que serán usadas y detalladas en capítulos posteriores:

1.4.2.1 TEMPERATURA

Los cultivos agrícolas, flores, plantas medicinales necesitan estar adaptadas a un cierto rango de temperatura para que puedan crecer y madurar de manera apropiada, por lo general la temperatura dentro de un invernadero dependiendo del día, varía desde los 10° centígrados como mínimo y a 30° como un máximo dependiendo del tipo de cultivo que se requiera [5]. En las noches se debe configurar al invernadero a temperaturas menores que son denominadas como temperaturas diurnas que varían entre los 5° y 7° centígrados con el fin que la temperatura del suelo sea la más beneficiosa para las plantas o cultivos [5]. La importancia de la temperatura y el diseño del invernadero influyen directamente con el proceso de la fotosíntesis que tienen las plantas o cultivos por lo que la calidad también estará afectada por estos cambios. [6]

1.4.2.2 Humedad

Dentro de un invernadero a medida que aumenta la temperatura exterior se retiene más humedad y se habla de la cantidad de vapor de agua que se concentra en el aire dentro del invernadero, también esta cantidad dependerá directamente de la estructura de la cuál este diseñada. La variable humedad ocupa un papel fundamental dentro de un invernadero debido a que pueden ocasionar fenómenos adversos en los cultivos y en las plantaciones como son las plagas o el proceso de crecimiento sea más lento de lo común. El nivel de la humedad está en un rango de 50% a un 80% y depende del tipo de cultivos que se tenga en el invernadero, es decir, si es para plantas medicinales, vegetales, flores [7].

1.4.2.3 Humedad relativa

La variable humedad relativa está ligada directamente con la humedad, esta variable se mide en porcentaje llegando a ser un nivel de aire saturado cuando se obtiene un valor de 100% y un nivel de aire seco cuando llega a 0%, es por esta razón que se busca los valores óptimos para que las plantaciones del invernadero. [8].

1.4.2.4 Nivel de CO₂

El nivel de CO₂ afecta directamente al proceso fotosintético de las plantas o cultivos que se tiene dentro del invernadero y depende de otros factores tales como, el tamaño del invernadero, el tipo de ventilación que se tenga, el número de cultivos o plantas y también de la manipulación del ser humano dentro del mismo [9]. Con un buen nivel de CO₂ permite un aumento del rendimiento, producción y calidad de la cosecha, además esta variable se mide en partes por millón (ppm), por lo general se requiere que el nivel se encuentre entre 800 y 1200 ppm [10].

1.4.2.5 Luminosidad

El nivel de luminosidad dentro de un invernadero es un factor fundamental, debido a la cantidad de luz que reciban las plantaciones o cultivos se puede optimizar el crecimiento, producción y calidad de estos. Además, esta cantidad de luz variará según la época del año que se tiene, si es invierno, verano, tormentas, etc. Por lo general se tiene un rango entre 10mil y 20mil luxes, aunque dependiendo de la plantación que se tenga este valor se puede incrementar en un rango de 20mil a 50mil luxes [11]. Para mantener el estado óptimo para el invernadero cuando haya épocas que no exista mucha luz natural, se deberá hacer uso de lámparas o iluminación artificial para compensar el nivel adecuado de los cultivos [12].

1.4.3 Arduino

Se puede definir al Arduino como una plataforma de código abierto, con el fin de diseñar distintos proyectos o prototipos con los distintos modelos que existen en el mercado, cómo UNO, DUE, NANO, YUM, LEONARDO, entre muchos otros. Cada modelo tiene distintos y potentes microcontroladores diseñados e integrados en sus placas que permiten la conexión de sensores, actuadores y varios componentes electrónicos juntamente con sus pines de entrada y salida. Arduino se complementa con el lenguaje de programación de Arduino denominado como Wiring y el Software Arduino o también llamado IDE [13].

Así como se tiene modelos desarrollados para satisfacer los requerimientos de cada usuario, existe de igual manera complementos para una sola placa sea mucho más poderosa como otorgándole Wifi, Ethernet, Bluetooth o una conexión con Linux, denominados “shields”. Para este proyecto se hizo uso de la plataforma Arduino UNO con el Dragino Yum shield.

1.4.3.1 ARDUINO UNO

La placa Arduino UNO es una de las más vendidas y utilizadas alrededor del mundo por sus amplias prestaciones debido a que cuenta con un poderoso microcontrolador ATmega328P, que es parte de la familia de microcontroladores AVR. Consta de 14 pines digitales de entrada y salidas I/O, de los cuales 6 pueden ser usados como PWM y otras 6 como entradas analógicas; a estas prestaciones se suma una velocidad de reloj de 16MHz, puerto USB, tiene comunicación UART, I2C, SPI, puerto de alimentación de 5V, botón de reinicio para recargar los scripts que son denominados “sketchs” [14].



Figura 1.1. Arduino UNO [14].

Adicionalmente, la placa Arduino UNO es compatible con una gran variedad de shields o placas de expansión que permite agregar funciones adicionales, como ya se mencionó anteriormente se utilizó el shield Dragino Yum debido a que la placa Yum durante el desarrollo del proyecto no estuvo disponible y el Shield Dragino Yum presentó las mismas funcionalidades y prestaciones.

1.4.3.2 Shield Arduino

Un shield también conocido como una placa de expansión, permite expandir las funcionalidades de hardware de una placa Arduino ordinaria, los shields se conectan directamente encima de una placa Arduino que sea compatible sin la necesidad de realizar conexiones o su vez soldar componentes. Entre las ventajas que ofrecen los shields se puede encontrar la compatibilidad, conexión directa, funcionalidades de Wi-Fi, Bluetooth, radio, etc. Para cada shield se tiene su propia librería debido a que de igual manera son de código abierto y puede ser ejecutado mediante el IDE de Arduino. Adicionalmente, la mayoría de shields están diseñados para operar con las placas Arduino UNO y Arduino Mega [15].

1.4.3.3 Dragino YUM Shield

El shield Dragino Yum permite tener una funcionalidad con un ambiente de Linux, permitiendo de esta manera crear proyectos para la automatización y para el presente proyecto se usó para dar seguimiento a las variables de un invernadero mediante una página web creada en Linux. Como se mencionó en el punto anterior los shields tienen compatibilidad para cada una de las placas Arduino, en la tabla 1 se puede observar con que placas y que funcionalidades extra ofrece [16]:

Tabla 1.1. Compatibilidad con distintas placas Arduino [16].

Placa	Programación USB	Programación Wifi	Programación Web	Consola Serial USB	Consola IDE Wifi	Consola SSH
101	SI	NO	NO	SI	SI	SI
DUE	SI	SI	SI	SI	SI	SI
Leonardo	SI	SI	SI	SI	SI	SI
Mega2560 Rev3	NO	SI	SI	NO	SI	SI
Uno Rev3	NO	SI	SI	NO	SI	SI
Zero	SI	SI	SI	SI	SI	SI

1.4.3.4 Arquitectura Dragino YUM Shield

El shield Dragino Yum se basa en un procesador Qualcomm Atheros AR9331, el cual permite la conexión con el sistema Linux basada en OpenWrt permitiendo la creación de una página web y servicios en la nube debido a que tiene el servicio de página web uHTTPd, la placa está conformada por una interfaz Wi-Fi, un puerto USB, botón de reinicio, puerto ethernet, una ranura microSD, 20 pines de entrada y salida, puede ser manipulado mediante web GUI, vía SSH, Wifi o por un puerto LAN. En la figura 1.2 se puede observar el esquemático del shield Dragino Yum [17].



Figura 1.2. Dragino Yum Shield [18].

Para el presente trabajo se hizo uso del shield Dragino Yum modelo v2.4 que presenta las siguientes características [17]:

- Bajo consumo de energía.
- Código abierto Linux (OpenWrt).
- Tiene un servidor web.

- El software se puede actualizar usando la red.
- Permite guardar proyector debido a que tiene una ranura microSD.
- Se puede acceder mediante Wi-Fi, cable LAN, mediante SSH o vía Web.
- Compatible con placas Arduino Uno, DUE, Mega.

Bajo estas características que ofrece el shield, el Arduino Uno provee de alimentación al Dragino Yum shield para poder usar todas las funcionalidades que fueron mencionadas previamente. En la figura 1.3 se puede observar el ensamblado entre el Arduino UNO con el shield Dragino:



Figura 1.3. Dragino Yum Shield + Arduino UNO [18].

Para finalizar, durante todo el documento que se presenta a continuación, se hace referencia al shield Dragino Yum + Arduino Uno, debido a que la placa Arduino Yum no estuvo disponible en la ciudad, pero las prestaciones que ofrece el shield Dragino Yum comparada con la placa Arduino Yum son las mismas y los aplicativos que se van a usar durante la resolución del proyecto son los mismos.

1.4.4 SERVIDOR WEB Y LINUX

Un servidor web es un programa o un software diseñado para presentar, alojar o almacenar datos a través de internet [19]. Para los servidores que son diseñados en Linux entregan contenido en HTML y se pueden ejecutar aplicaciones mediante PHP, Python, Ruby y entre otras. Para crear un servidor web se los puede levantar en diferentes entornos, como, Apache, uHTTPd, Nginx, LiteSpeed, entre otros. Para el Dragino Yum Shield ya viene implementado un servidor web denominado uHTTPd, debido a que la placa tiene una distribución OpenWRT y con esto el paquete del servidor web preinstalado [32] por lo cual, para presentar los valores de las variables comentadas previamente, se hace uso de este servicio y el cual se detalla de mejor manera más adelante [20] [21].

1.4.5 HTLM, CSS Y JAVASCRIPT

1.4.5.1 HTLM (Hypertxt Markup Lenguaje)

Es un tipo de lenguaje de programación que permite diseñar en texto “plano” el contenido de las páginas web, es decir, hace referencia para la estructura o esqueleto de la página web para presentar tablas, imágenes, diseños, enlaces, etc. Para el presente proyecto se hizo uso de dos archivos html, el primero index.html que es la caratula del proyecto y el segundo archivo datos.html que es el diseño de toda la página web [22]; que con la ayuda del archivo en formato CSS y otro con JavaScript permitirá que la página web sea más interactiva con el usuario.

1.4.5.2 CSS (Cascading Style Sheets)

Es un lenguaje que se usa para dar diseño, formato y estilos a los componentes que fueron diseñados en HTML de una página web, es decir, con el formato CSS permite definir cómo se presentará al usuario la página web final [23].

1.4.5.3 JavaScript

Con el lenguaje de programación de JavaScript se es capaz de implementar otras características a la página web, como adquisición de una base de datos exportadas de fuentes externas, juegos, animación y sitios web dinámicos [24]. Para el presente documento se hizo uso para leer la base de datos de las variables obtenidas y presentarla en la página web.

2 METODOLOGÍA

El presente trabajo hace énfasis en la creación, diseño y presentación de una página web sobre la placa Arduino Yum, la cual presentará de manera intuitiva e interactiva las variables de entorno que influyen dentro de un invernadero las cuales fueron comentadas previamente. Para levantar la página web se siguió un lineamiento y se hizo uso de diferentes plataformas que, trabajando en conjunto, permite la visualización de las variables en una página web diseñada en un entorno Linux con la placa Dragino Yum shield.

Como primer punto se tiene la configuración adecuada de la placa Arduino UNO con el shield Dragino Yum, debido a que la placa original, Arduino Yum, no fue posible conseguir dentro del país, pero esto es irrelevante para el desarrollo del presente proyecto debido a que el shield tiene las mismas prestaciones que la placa y como se había comentó en puntos anteriores, el Arduino Uno permite la alimentación a dicho shield, con lo cual es una placa viable para el presente documento.

Una vez se tiene la placa trabajando en la red deseada, se procede a realizar un bosquejo, o esqueleto de la página web la cual será diseñada con tres tipos de lenguaje de programación para páginas web, los cuales son HTML, CSS y JavaScript. Esto es posible debido a que el shield Dragino Yum ya tiene integrada a breves rasgos un servidor web denominado uHTTPd con lo cual creando archivos con extensiones .HTML, .CSS o, JS en el directorio /www” se puede levantar sin ningún problema la página web deseada. Al ser HTML un lenguaje de programación de “texto plano” y con su tipo de sintaxis que se hará referencia posteriormente, además que el editor de texto de Linux (vi o nano) tiene muchas limitaciones se hizo uso de un editor externo llamado Visual Studio Code que permite en sus funcionalidades mostrar en tiempo real el diseño de la página web a crear.

Para comprobar la conexión que existe entre la placa Dragino Yum Shield y el Arduino UNO, es decir, el envío de datos se hace uso de la librería bridge, que permite la interconexión entre placas y poder mostrar los valores de las variables a ser visualizadas en el monitor serial que ofrece el IDE de Arduino.

Teniendo la página web a diseñar se procede a obtener los datos de las variables del invernadero, temperatura, humedad, humedad relativa, nivel de CO2 y luminosidad. Para este punto se hizo uso del IDE de Arduino y qué librerías permitieron la interconexión entre placas y así tener una simulación con valores y rangos de estas variables aleatoriamente que podrán ser observados en la página web.

Como último y ya mencionado previamente, se detalla más a fondo los últimos detalles de la página web de acuerdo con los datos obtenidos, mostrando gráficas de cada una de las variables, la tabla que se exportó de las variables en los rangos y horas predefinidas.

Todos estos puntos o lineamientos que se divide este proyecto se detallarán posteriormente y se ahondará en los puntos más importantes que permitieron la realización y finalización de la página web, además de presentar los respectivos, códigos diseñados.

2.1 Configuración del shield Dragino Yum

Como se mencionó en puntos anteriores una de las ventajas que se tiene con el shield Dragino Yum es la posibilidad de conectarse a internet mediante Ethernet o mediante Wi-Fi además que fue diseñado para el internet de las cosas (IoT) y cuenta con la configuración de servicios web mediante Linux.

Al conectar por primera vez el shield con la placa de Arduino UNO, se debe de esperar que el led de color azul que indica la WLAN se prenda, una vez se pueda observar el color azul, en la computadora local donde se esté trabajando aparecerá la red Wi-Fi denominada

“Dragino” y para acceder a las configuraciones la IP por defecto es 192.168.240.1 con la contraseña por defecto “dragino” [18]. Una vez realizado el ingreso correctamente, se debe de dirigir al botón “system” y se procede a configurar contraseña de acceso, la red Wi-Fi a la que se requiere conectar como se puede apreciar en la figura 2.1.

The image shows the web control panel for a Dragino Yun device. It features a blue header with the Dragino logo and the text "Welcome to your Dragino Yun. Please enter password to access the web control panel". Below this is a login section with a "PASSWORD" field containing "dragino" and a "LOG IN" button. A message below the login field says "Please be sure you have cookies enabled before proceeding." The main configuration area is divided into two sections: "YÚN BOARD CONFIGURATION" and "WIRELESS PARAMETERS". The "YÚN BOARD CONFIGURATION" section includes fields for "YÚN NAME *" (InvernaderoEPN), "PASSWORD" (masked with dots), "CONFIRM PASSWORD" (masked with dots), and "TIMEZONE *" (America/Guayaquil). The "WIRELESS PARAMETERS" section includes a checkbox for "CONFIGURE A WIRELESS NETWORK" (checked), a "DETECTED WIRELESS NETWORKS" dropdown menu (Select a wifi network...), a "Refresh" link, a "WIRELESS NAME *" field (NETLIFE-uiodmchisir1), a "SECURITY" dropdown menu (WPA2), and a "PASSWORD *" field (masked with dots). At the bottom, there are two buttons: "DISCARD" and "CONFIGURE & RESTART".

Figura 2.1. Configuración de la red en el shield Dragino Yum.

Una vez se haya conectado la placa con la red Wi-Fi requerida, la IP del shield estará dentro de un pool de direcciones según el protocolo DHCP que se tenga en el equipo. Si se desconoce la dirección IP una vez se haya reiniciado la tarjeta se puede hacer uso de un programa llamado “Advanced Port Scanner”, el cuál una vez se proceda a buscar la dirección IP, se mostrará en la lista la dirección requerida para poder acceder vía SSH a la línea de comandos y configurar los archivos pertinentes para la creación de la página web. Para el presente caso se puede observar la configuración del programa en la figura 1.5.

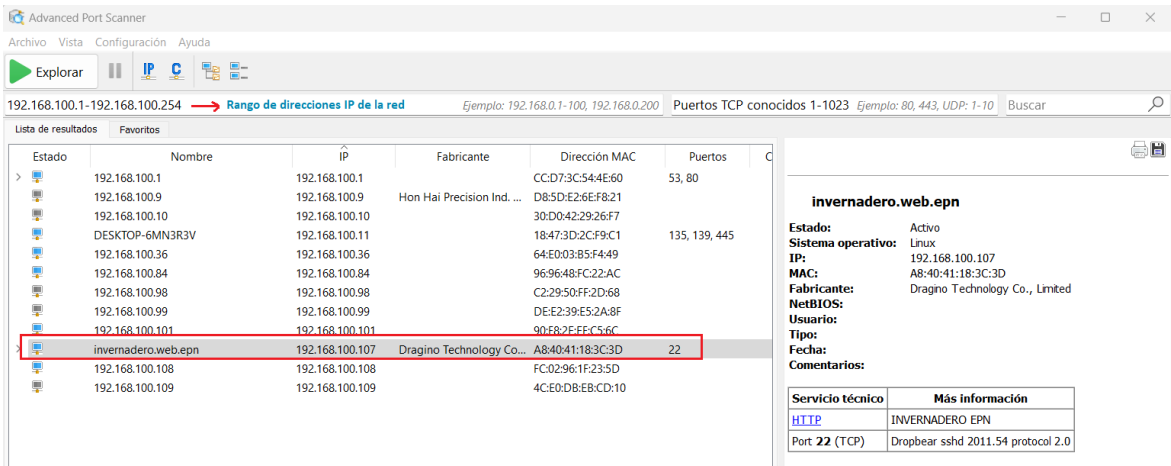


Figura 2.2. IP para el servidor Web.

2.1.1 ACCESO VÍA SSH

Al tener la dirección IP se puede comenzar a usar las funcionalidades de Linux que ofrece el shield Dragino Yum, para esto se debe de acceder vía SSH mediante algún programa que puede ser: PuTTY, XShell, MobaXterm, etc. Para el presente documento se utilizó el programa MobaXterm que permite crear una sesión SSH y tiene una interfaz amigable con el usuario [25]. Para acceder se debe de crear un cliente SSH con la dirección IP y con el usuario "root" una vez dentro se debe de ingresar la contraseña que fue establecida al momento de configurar por primera vez el shield o si no se cambió la contraseña, se debe de mantener la contraseña por defecto que es "dragino". Una vez dentro de la placa se presentará la siguiente información como se puede apreciar en la figura 2.3.

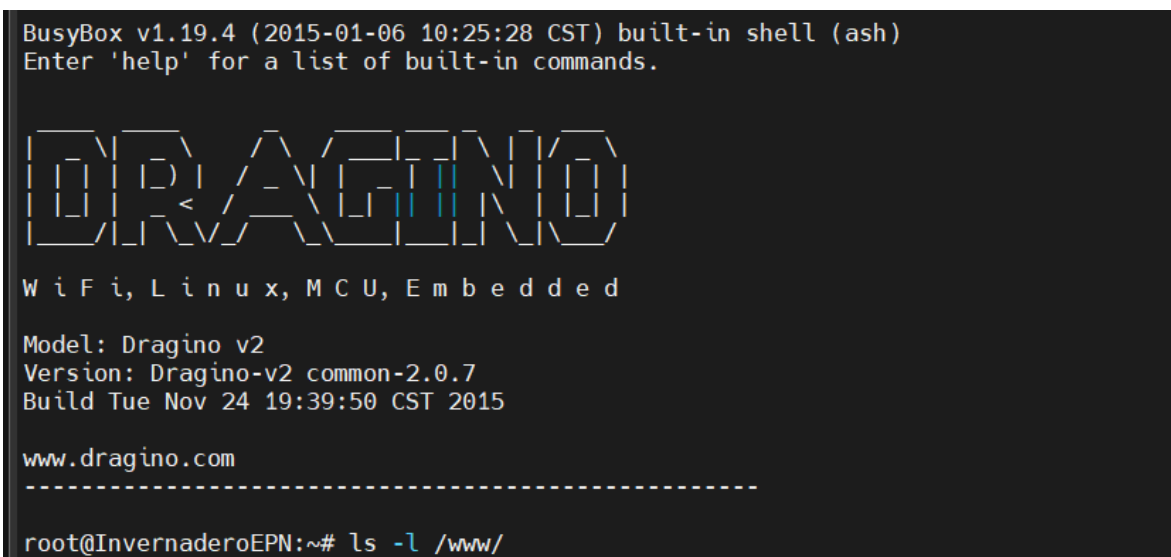


Figura 2.3. Conexión SSH de la placa Dragino YUM Shield.

2.2 Creación de los archivos de la para la página web

El shield Dragino Yum al trabajar con el procesador Atheros AR9331 permite que se tenga conectividad Wi-Fi o Ethernet y ejecutar funcionalidades o programas dentro de un ambiente de Linux [16]. Como se había comentado previamente la versión de Linux que se tiene dentro de la placa tiene una ligera configuración de un servidor web uHTTPd, por lo que es posible la creación de una página y un servidor web. Para el presente caso se crearon los siguientes archivos para la creación de la página web en el directorio /www/.

1. Archivo “index.html” en el cual se contiene la portada del proyecto
2. Archivo “datos.html” en el cual se presenta todas las variables a ser monitoreadas en el invernadero, además que muestra los valores máximos, mínimos y promedio de la base de datos obtenida.
3. Archivo “styles.css” el cual permite dar diseño a los archivos con extensión .HTML, es decir, tamaño de fuente, color, disposición, etc.
4. Archivos en extensión .JS, que permiten graficar cada una de las variables que serán presentadas en las páginas diseñadas en HTML, las cuales, se hablarán posteriormente.
5. Archivos en extensión .HTML para cada una de las variables para mostrar las gráficas y fecha que fueron capturados los datos.
6. Archivo “historico.html” en el cual se presenta toda la tabla de la base de datos obtenido con la posibilidad de descargarse en formato CSV.
7. Archivo “script.js” el cual está diseñado con JavaScript que permite obtener los valores máximos, mínimos y promedios de cada una de las variables.

Cada uno de los archivos expuestos anteriormente serán ahondados de mejor manera y con más detalles a lo largo de este documento. En este punto se hablará de las principales características que se tiene en los lenguajes HTML, CSS y JavaScript y la sintaxis que se tiene para cada uno de estos que dieron la posibilidad del diseño de la página web.

2.2.1 DIRECTORIO /WWW/

Una de las facilidades que ofrece la placa Dragino Yum Shield, es el servicio de una página web, debido a que el sistema operativo Linux viene integrado un ligero servicio denominado uHTTPd. Es por esto por lo que se puede levantar directamente la página web en el directorio /www/ debido a que es el directorio raíz del servidor web uHTTPd [33]. Todos los

archivos HTML, CSS, JavaScript, imágenes y base datos en formato Excel o CSV deben de ser cargados y subidos en este directorio. Para que sea procesado y se pueda mostrar la página web requerida. En la siguiente figura se puede observar el directorio /www/ actual y los archivos que fueron usados para levantar la página web y que serán detallados posteriormente.

```
root@InvernaderoEPN:/www# ls -l
-rw-r--r-- 1 root root 28056 Aug 23 00:45 Logo.png
-rw-r--r-- 1 root root 3785 Aug 23 00:45 basedatos.csv
drwxr-xr-x 2 root root 27 Nov 24 2015 cgi-bin
-rw-r--r-- 1 root root 3124 Aug 23 00:43 datos.html
-rw-rw-r-- 1 root root 1150 Nov 22 2015 favicon.ico
-rw-r--r-- 1 root root 277442 Aug 23 00:45 foto.webp
-rw-r--r-- 1 root root 1376 Aug 23 00:43 grafica_co2.js
-rw-r--r-- 1 root root 1436 Aug 23 00:43 grafica_humedadrel.js
-rw-r--r-- 1 root root 1388 Aug 23 00:43 grafica_luminosidad.js
-rw-r--r-- 1 root root 3180 Aug 23 00:57 historico.html
-rw-r--r-- 1 root root 1025 Aug 23 00:49 humedad.html
-rw-r--r-- 1 root root 1407 Aug 23 00:43 humedad_grafica.js
-rw-r--r-- 1 root root 1034 Aug 23 00:51 humedadrel.html
-rw-r--r-- 1 root root 881 Aug 23 00:43 index.html
drwxrwxr-x 2 root root 53 Nov 24 2015 keystore_manager_example
drwxr-xr-x 5 root root 68 Nov 24 2015 luci-static
-rw-r--r-- 1 root root 1019 Aug 23 00:52 luminosidad.html
-rw-r--r-- 1 root root 1021 Aug 23 00:51 nivelco2.html
-rw-r--r-- 1 root root 4077 Aug 23 00:43 script.js
-rw-r--r-- 1 root root 3346 Aug 23 00:45 styles.css
-rw-r--r-- 1 root root 1042 Aug 23 00:50 temperatura.html
-rw-r--r-- 1 root root 1673 Aug 23 00:59 temperatura_grafica.js
root@InvernaderoEPN:/www#
```

Figura 2.4. Archivos requeridos para la página web dentro del directorio /www/.

2.2.2 HTML

El Lenguaje de Marcado de Hipertexto, ampliamente conocido como HTML por su abreviación, es un lenguaje de marcado que permite dar esqueleto a una página web [22]. En este tipo lenguaje se hace uso de etiquetas las cuales permiten mostrar una palabra, imagen, tamaño de fuente, vínculos, etc. Las tres principales partes que conforman un pequeño código en HTML son: Etiqueta de apertura, que se conforma del tipo de elemento dentro de paréntesis angulares < >, el contenido que tendrá la etiqueta y por último la etiqueta de cierre con el tipo de elemento dentro de </>. Cabe mencionar que también se puede tener etiquetas anidadas según las necesidades que se requiera. A continuación, se indican las principales etiquetas de un archivo HTML.

Tabla 2.1. Etiquetas comunes en HTML [26][27].

Etiqueta	Definición
<!DOCTYPE html>	Etiqueta para especificar la versión de HTML que se va a utilizar en la página
<html > ... </html>	Etiqueta para definir el inicio y el final de la página web
<title>	Etiqueta que indica el nombre que tendrá la pestaña en el navegador web.
<body> ... </body>	Etiqueta que indica la sección de todo el documento, dentro de esta etiqueta se incluyen todos los elementos que serán visualizados en la página web.
<h1> a <h6>	Etiqueta que definen el título o encabezado o las jerarquías en la página web. Teniendo a h1 como título 1 y a h6 como título 6
<a>	Etiqueta para definir un enlace externo o a otra sección de la página
<p> ... </p>	Etiqueta que define un párrafo.
 ... 	Etiqueta para dar formato de negrita a un párrafo o un texto.
 ... 	Etiqueta para dar formato de cursiva a un párrafo o un texto.
<u> ... </u>	Etiqueta que permite dar el formato de subrayado de un párrafo o un texto.
	Etiqueta para subir y mostrar una imagen en la página web.
<div> ... </div>	Etiqueta que permite crear una división o un contenedor.
<tr> </tr>	Etiqueta que indica una fila de celda de una tabla
<td> </td>	Etiqueta que indica una celda de una tabla.
<th> </th>	Etiqueta para definir el encabezado de una celda

Un ejemplo sencillo realizado en HTML se lo puede observar en el Código 2.1, en el cual se ocupan algunas de las etiquetas mencionadas en la tabla anterior y con su respectivo comentario de que es lo que hace cada una de las líneas de código:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo HTML</title>
</head>
<body>
  <h1>Encabezado 1</h1>
  <h2>Encabezado 2</h2>
  <p><strong>Esto es un párrafo con texto en negrita.</strong></p>
  <p><em>Esto es un párrafo con texto en cursiva.</em></p>
  <p><u>Esto es un párrafo subrayado.</u></p>
</body>
</html>
```

Código. 2.1. Código HTML Básico [autor].

Una de las limitaciones que se tiene con el lenguaje HTML es no poder dar un diseño entero a la página web, solamente permite diseñar el “esqueleto” de la página web que se requiera diseñar, es por esta razón que se complementa con otros lenguajes como es CSS, el cual, permite dar diseños o estilos a la página web a las etiquetas que fueron descritas en la tabla 2.1 y también se complementa con el lenguaje de JavaScript, que permite mayor interactividad, manejo de archivos y muchas más funcionalidades dependiendo de lo que se requiere mostrar en la página web [22].

Para el presente documento se hizo uso de ocho archivos con extensión .HTML para que la página a implementar sea funcional y amigable con el usuario cumpliendo de esta manera las expectativas del presente componente.

En la tabla 2.2 que se muestra a continuación, se detalla y se explica brevemente lo que cada archivo presenta al ingresar a las distintas páginas web.

Tabla 2.2 Descripción de archivos HTML usados en el proyecto.

Nombre del archivo	Descripción
index.html	Presenta la carátula o portada de la página web.
datos.html	Dentro de esta página web, se encuentran varios botones interactivos con cada una de las variables a ser monitoreadas, además que se presenta el valor máximo, mínimo y el promedio de cada una de estas variables. Otra funcionalidad que presenta la página es redirigir a las páginas webs pertinentes de cada una de las variables para lograr observar y monitorear su gráfica; por último se tiene el botón que redirige al histórico o la tabla completa de las variables.
temperatura.html	En esta página se presenta la gráfica de los valores obtenidos con respecto a la variable temperatura, en base a la hora y fecha de cada valor capturado.
humedad.html	En esta página se presenta la gráfica de los valores obtenidos con respecto a la variable humedad, en base a la hora y fecha de cada valor capturado.
humedadrel.html	En esta página se presenta la gráfica de los valores obtenidos con respecto a la variable humedad relativa en base a la hora y fecha de cada valor capturado.
nivelco2.html	En esta página se presenta la gráfica de los valores obtenidos con respecto a la variable nivel de CO ₂ , en base a la hora y fecha de cada valor capturado.
luminosidad.html	En esta página se presenta la gráfica de los valores obtenidos con respecto a la variable luminosidad, en base a la hora y fecha de cada valor capturado.
historico.html	En esta página se presenta la tabla general de todos los datos capturados, además que tiene la funcionalidad de descargarse la tabla en formato CSV.

El contenido de cada uno de estos archivos se puede encontrar en Anexo I, Anexo II, Anexo III, Anexo IV, Anexo V, Anexo VI, Anexo VII, Anexo VIII, respectivamente.

2.2.3 CSS

El lenguaje de hojas de estilo, CSS (Cascading Style Sheets) o por su traducción al español Hojas de Estilo en Cascada, permite realizar estilos a los archivos o documentos HTML, color, tamaño de fuente, secciones, posicionamiento y muchos más estilos que se requiera en la página web [23]. La sintaxis o forma del código CSS se basa en una “regla” y consta de lo siguiente:

- **Selector:** selecciona una etiqueta del archivo HTML que puede ser <p> o <h1> y se aplica el estilo que se requiera, es decir, tamaño de fuente, color, posicionamiento. Para llamar a la etiqueta puede ser mediante un ID o una clase, por lo general se hace uso de clase para identificar de mejor manera en el cuerpo del archivo HTML.
- **Propiedad:** es la forma en la cual se presentan los elementos seleccionados, es decir, su apariencia y diseño. Dentro de las propiedades puede modificarse el color, texto, fondo, tamaño, etc.
- **Declaración:** especifica la regla, estilo o valor para presentar elementos.
- **Valor a la propiedad:** luego de definir alguna propiedad, debe ser terminada en dos puntos y elegir el estilo que se desea.

A continuación, se puede apreciar un ejemplo sencillo de aplicar formato a las etiquetas de un archivo HTML:

```
.Historial{
    text-align: center;
}
.item{
    font-size: 70px;
}
.humedad {
    background-color: #DD5A7A; /* Color para humedad */
}
```

Código. 2.2 Código básico para dar formato a una etiqueta con una clase en CSS [autor].

Se tiene varios estilos y diseños que pueden ser aplicados para dar una mejor presentación a la página web, en la tabla 2.3 se presentan alguno de los diseños que pueden ser usados:

Tabla 2.3. Descripción de algunas propiedades usadas en CSS [28].

Propiedades	Descripción
.class	Selección de elementos con el nombre de la clase.
#id	Selección del elemento con el ID.
px	Tamaño del pixel.
color	Establece un valor de texto que puede ser colocado en rgb o en formato hexadecimal.
font-size	Establece tamaño del texto.
background-color	Especifica el color de fondo de un elemento.
position	Especifica el esquema de posicionamiento.
padding	Establece el espacio entre el contenido y el borde de un elemento.
text-align	Establece la alineación horizontal del texto de un elemento.
border-radius	Establece el radio de las esquinas de un elemento

Una vez entendido acerca del uso y las posibilidades que ofrece CSS, en el presente documento se hizo uso de un solo archivo llamado “styles.css” dentro del cual, contiene todo el diseño de la página web, color de fondo, imágenes, diseño de botones, color de texto y muchas cosas adicionales que pueden ser observadas en el Anexo IX.

2.2.4 JAVASCRIPT

El lenguaje de programación JavaScript, ofrece una gran variedad de funcionalidades y características interactivas en un sitio web, puede ser: gráficas, control de documentos, juegos, animación, entre otros. JavaScript ofrece un mundo de posibilidades para el diseño y funcionalidades para una página web, para el presente proyecto se hizo uso solo un fragmento que ofrece JavaScript y es para monitorear las variables en forma de gráficas [24]. Con la ayuda de librerías que serán llamadas en los archivos HTML y archivos con extensión .js, permite visualizar de manera interactiva las variables del invernadero.

Para el presente caso, la visualización de las variables mediante gráficas se hace uso de la librería CharJS. Para cargar o incluir un código de JavaScript se debe de utilizar la etiqueta <script> sobre el archivo HTML que se requiera las funcionalidades, dentro de la etiqueta <head> del mismo archivo. Para el procedimiento de carga y uso de esta librería se tiene lo siguiente [29]:

- Dentro del archivo HTML se hace uso de la librería chart.js según la siguiente línea de código:

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

Código. 2.3. Llamado a la librería chart.js para gráficas.

- Se debe de añadir un elemento denominado canvas en el archivo HTML, con un ID específico para acceder y dar diseño a la gráfica.

```
< canvas id="Temperatura"></canvas>
```

Código. 2.4. Etiqueta canvas para graficar la variable temperatura.

- Por último, se comienza a programar para que se pueda crear el grafico y poder visualizarlo en el archivo HTML. Vale mencionar que para dar formatos extras se hace uso de CSS [28].

```
const temperatureCtx =
document.getElementById('temperatureChart').getContext('2d');

    new Chart(temperatureCtx, {
      type: 'line',
      data: {
        labels: labels,
        datasets: [{
          label: 'Temperatura (°C)',
          data: temperatureData,
          borderColor: 'rgba(255, 99, 132, 1)',
          backgroundColor: 'rgba(255, 99, 132, 0.2)',
          borderWidth: 1
        }]
      },
      options: {
        scales: {
          y: {
            beginAtZero: false
          }
        }
      }
    });
```

Código. 2.5. Fragmento de código para graficar la variable temperatura en JavaScript.

Para la visualización de todas las variables a ser monitoreadas en el invernadero gráficamente, se hizo uso de seis archivos programados en JavaScript. En la tabla 5 se puede observar el nombre y una pequeña descripción de cada uno:

Tabla 2.4. Descripción de los archivos .JS.

Nombre del archivo	Descripción
temperatura_grafica.js	Permite la visualización de la gráfica de la variable temperatura.
humedad_grafica.js	Permite la visualización de la gráfica de la variable humedad.
grafica_humedadrel.js	Permite la visualización de la gráfica de la variable humedad relativa.
grafica_co2.js	Permite la visualización de la gráfica de la variable nivel de CO2.
grafica_luminosidad.js	Permite la visualización de la gráfica de la variable luminosidad.
script.js	Permite la visualización de los valores máximo, mínimo y promedio de cada variable.

El contenido de los códigos previamente mencionados se puede visualizar en Anexo X, Anexo XI, Anexo XII, Anexo XIII, Anexo XIV, Anexo XV respectivamente.

2.3 Base de datos

Los datos “simulados” a ser monitoreados fueron obtenidos mediante una hoja de Excel, en el cuál consta las variables de mayor atención, como son: temperatura, humedad, humedad relativa, nivel de co2 y luminosidad. Cada uno de los valores obtenidos fueron monitoreados para el presente documento durante el mes de julio con un periodo de cada hora. Los valores para graficar se podrán ver en el apartado siguiente, en resultados. Vale mencionar que esta simulación de valores ya fue validada en otro componente, con lo cual solo se presentan estos resultados mediante el despliegue de una página web.

En la tabla 2.5 presentada a continuación se puede observar el rango de valores de las variables a ser monitoreadas, estos valores fueron considerados de acuerdo con el promedio que se tienen los invernaderos convencionales:

Tabla 2.5. Rango de valores a ser monitoreados.

VARIABLE	Rango
Temperatura	10°C – 30°C
Humedad	50% - 80%
Humedad Relativa	40% - 60%
Nivel de CO2	800ppm – 1200ppm
Luminosidad	10000luxes – 50000luxes

2.4 Conexión con el IDE de Arduino

Al usar la placa de Arduino UNO con el Dragino Yum Shield y poder interconectar las placas mediante las funcionalidades que ofrece cada placa o a su vez ejecutar programas sobre el IDE de Arduino denominado como “sketch” primero se debe de realizar operaciones previas, como descarga de librerías y actualización de las placas o tarjetas sobre el IDE de Arduino.

Una vez configurado la placa Dragino Yum Shield a la red Wi-Fi que se va a trabajar, tema que fue explicado y detallado en puntos previos. Se procede a configurar el IDE de Arduino de tal forma que reconozca a las placas Arduino UNO y Dragino Yum Shield. A diferencia de conectar las placas con su respectivo cable USB a Arduino y que reconozca el puerto COM, al estar conectado mediante Wi-Fi no hay la necesidad de hacer uso de cables físicos y al estar la placa en la misma red los programas o sketch son cargados mediante la red Wi-Fi configurada [30].

Dentro del IDE de Arduino se deben de realizar las siguientes configuraciones:

1. Descargar el paquete de todas las tarjetas disponibles, para esto se debe de dirigir a “Herramientas -> Placa -> Gestor de Placas” y descargar la librería “Arduino AVR Boards”. Para versiones anteriores se puede encontrar como “Dragino Yum”, para las versiones más actuales como shield Dragino Yum, el cual, es reconocido como la placa Arduino Yum.

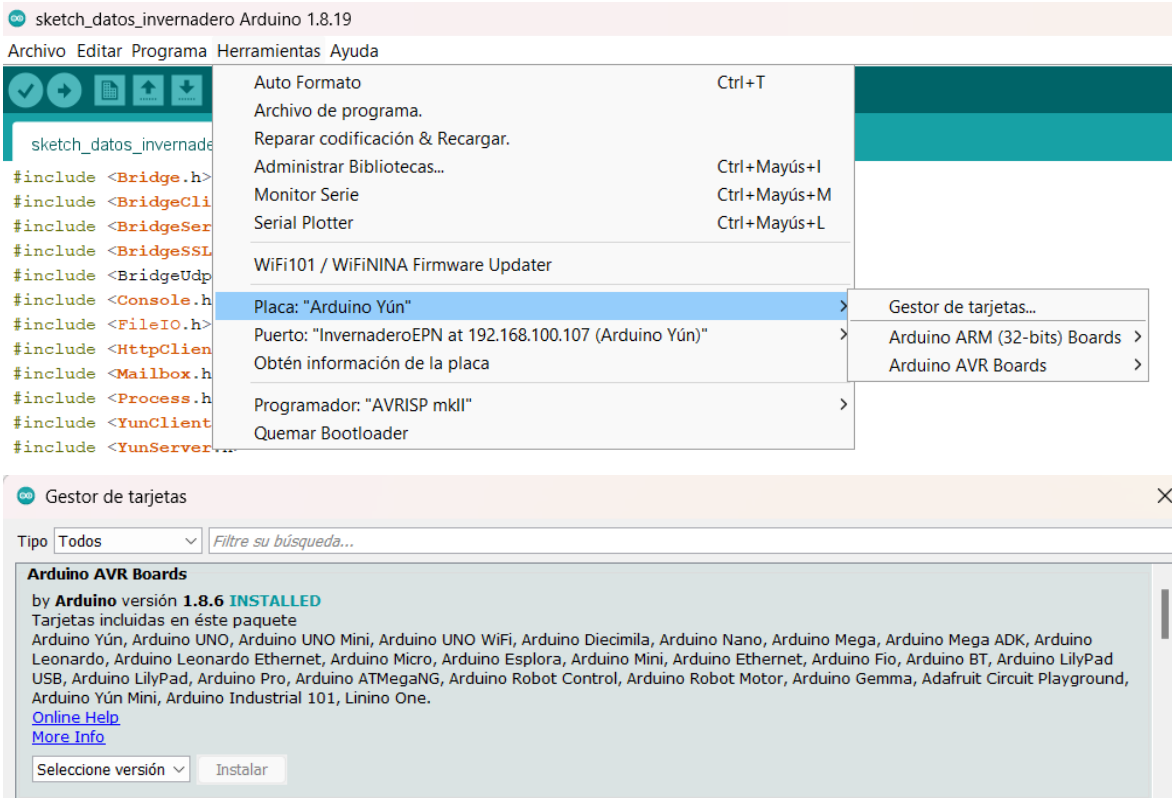


Figura 2.5. Instalación de tarjetas de Arduino [autor].

- Una vez señalado la placa "Arduino Yum" se debe de seleccionar el puerto, como ya se mencionó previamente al estar conectado mediante una red Wi-Fi no se requiere hacer uso de cables USB, si no buscar la red en la que se encuentra la placa y seleccionarlo.

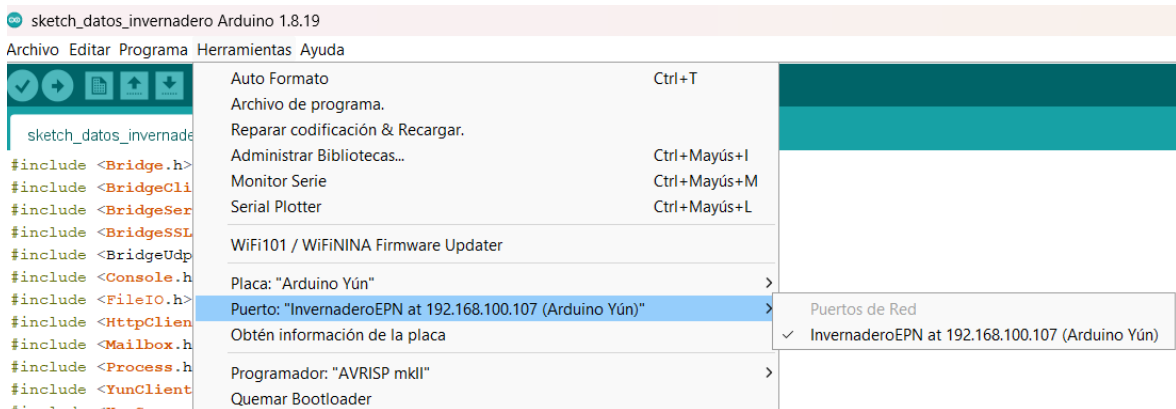
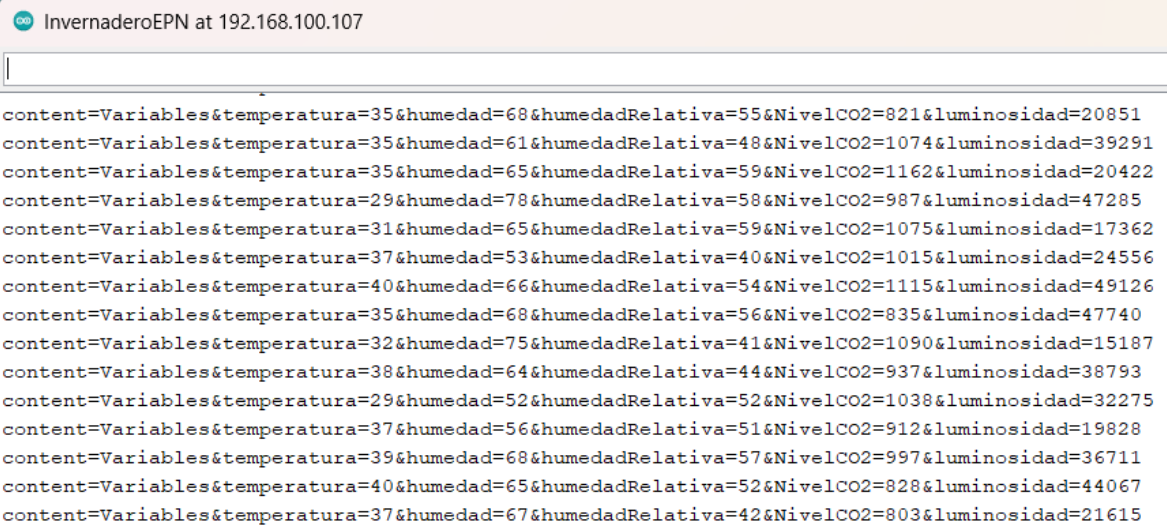


Figura 2.6. Selección del puerto que se encuentra la placa Arduino Uno + Dragino Yum Shield [autor].

Una vez realizado los pasos anteriores, nuestro IDE esta en condiciones para trabajar con la placa conformada con el Arduino UNO con el Dragino Yum Shield. Para que exista una conexión entre el IDE de Arduino con el servidor web diseñado se hace uso de la librería bridge. La librería bridge permite la comunicación entre el procesador Linux con el

Microcontrolador de la placa a usarse, es este caso con el microcontrolador de la placa Arduino UNO. Entre las funcionalidades que ofrece esta librería se encuentra, almacenamiento compartido, ejecución de procesos de Linux, servidores HTTP entre otras.

Para la conexión entre el Arduino Uno con el servidor web se lo hace a través de la librería “bridge”, debido a que facilita la comunicación entre el procesador del microcontrolador y el sistema operativo Linux que se tiene en la placa [34]. Se puede usar juntamente con la función “console” para presentar un mensaje por el monitor serial. En la figura presentada a continuación se puede observar el envío de datos usando la librería bridge para ser capturados y presentados en la página web.



```
content=Variables&temperatura=35&humedad=68&humedadRelativa=55&NivelCO2=821&luminosidad=20851
content=Variables&temperatura=35&humedad=61&humedadRelativa=48&NivelCO2=1074&luminosidad=39291
content=Variables&temperatura=35&humedad=65&humedadRelativa=59&NivelCO2=1162&luminosidad=20422
content=Variables&temperatura=29&humedad=78&humedadRelativa=58&NivelCO2=987&luminosidad=47285
content=Variables&temperatura=31&humedad=65&humedadRelativa=59&NivelCO2=1075&luminosidad=17362
content=Variables&temperatura=37&humedad=53&humedadRelativa=40&NivelCO2=1015&luminosidad=24556
content=Variables&temperatura=40&humedad=66&humedadRelativa=54&NivelCO2=1115&luminosidad=49126
content=Variables&temperatura=35&humedad=68&humedadRelativa=56&NivelCO2=835&luminosidad=47740
content=Variables&temperatura=32&humedad=75&humedadRelativa=41&NivelCO2=1090&luminosidad=15187
content=Variables&temperatura=38&humedad=64&humedadRelativa=44&NivelCO2=937&luminosidad=38793
content=Variables&temperatura=29&humedad=52&humedadRelativa=52&NivelCO2=1038&luminosidad=32275
content=Variables&temperatura=37&humedad=56&humedadRelativa=51&NivelCO2=912&luminosidad=19828
content=Variables&temperatura=39&humedad=68&humedadRelativa=57&NivelCO2=997&luminosidad=36711
content=Variables&temperatura=40&humedad=65&humedadRelativa=52&NivelCO2=828&luminosidad=44067
content=Variables&temperatura=37&humedad=67&humedadRelativa=42&NivelCO2=803&luminosidad=21615
```

Figura 2.7. Envío de valores de las variables a través del IDE de Arduino.

El código diseñado que se usó en el IDE de Arduino enviar valores al servidor web, se puede observar en Anexo XVI.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

La culminación del presente proyecto fue satisfactoria, se logró diseñar una página web funcional, donde se puede dar seguimiento a las variables de entorno que involucra un invernadero. Usando los servicios que ofrece la placa de una manera intuitiva y amigable con el usuario.

3.1.1 VISUALIZACIÓN DEL SERVIDOR WEB

Para ingresar a la página web, se puede ingresar mediante la IP configurada previamente “192.168.100.107” o mediante su dominio “http://www.invernadero.web.epn.com/”. Como

primera parte de la presentación de resultados, se tiene la portada; que hace referencia al nombre del proyecto, autor y tutor.

Cuenta con un botón que una vez presionado, está configurado de tal manera que autodirigirá a otra ventana y presentará las variables a ser monitoreadas.



Figura 3.1. Portada de la página web.

Una vez presionado el botón “Ingresar” se despliega una ventana que muestra todas las variables con su valor máximo, mínimo y el promedio según se tenga en la base de datos. Además, una vez que se presiona el icono de la variable que se desea monitorear, se autodirigirá a otra ventana donde se podrán observar la gráfica obtenida de acuerdo con valores capturados en el periodo definido previamente.

Otra funcionalidad que tiene en esta ventana es la posibilidad de acceder al histórico, es decir, la tabla donde se muestran todos los valores obtenidos. En la figura 3.2, se puede observar de mejor manera el diseño comentado.

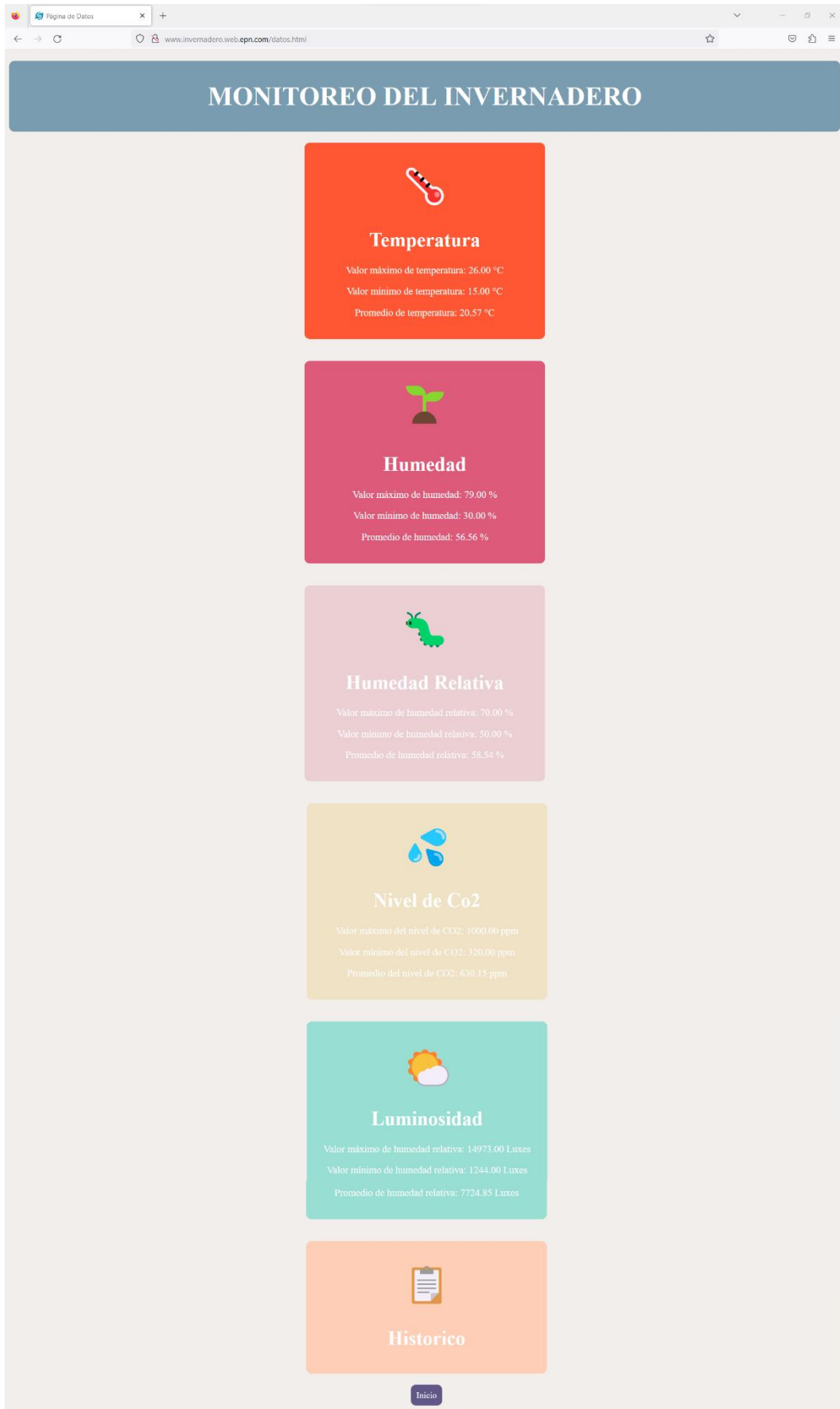


Figura 3.2. Visualización de las variables del invernadero.

Como se mencionó anteriormente, una vez que se selecciona la variable que se desea visualizar, se desplegará las gráficas pertinentes de las variables del invernadero.

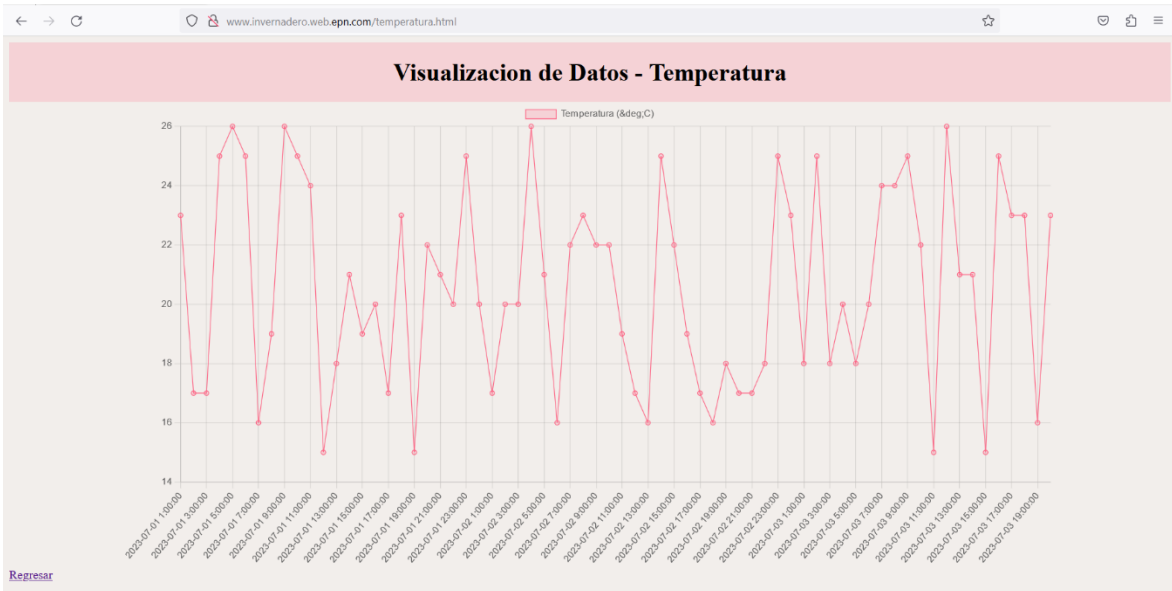


Figura 3.3. Gráfica de la variable temperatura.

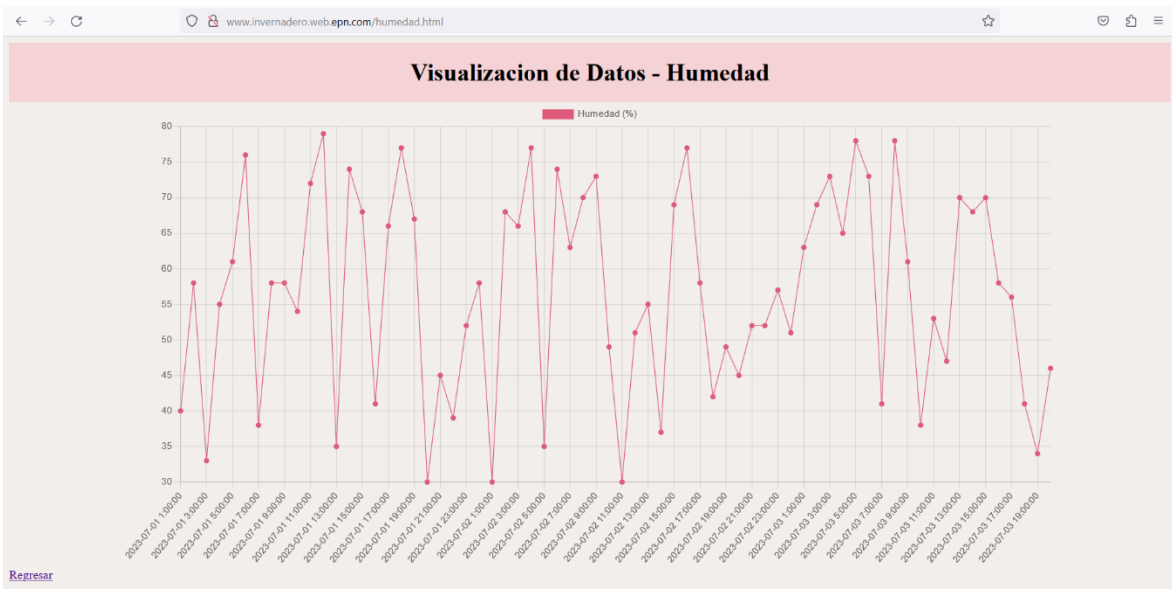


Figura 3.4. Gráfica de la variable humedad.

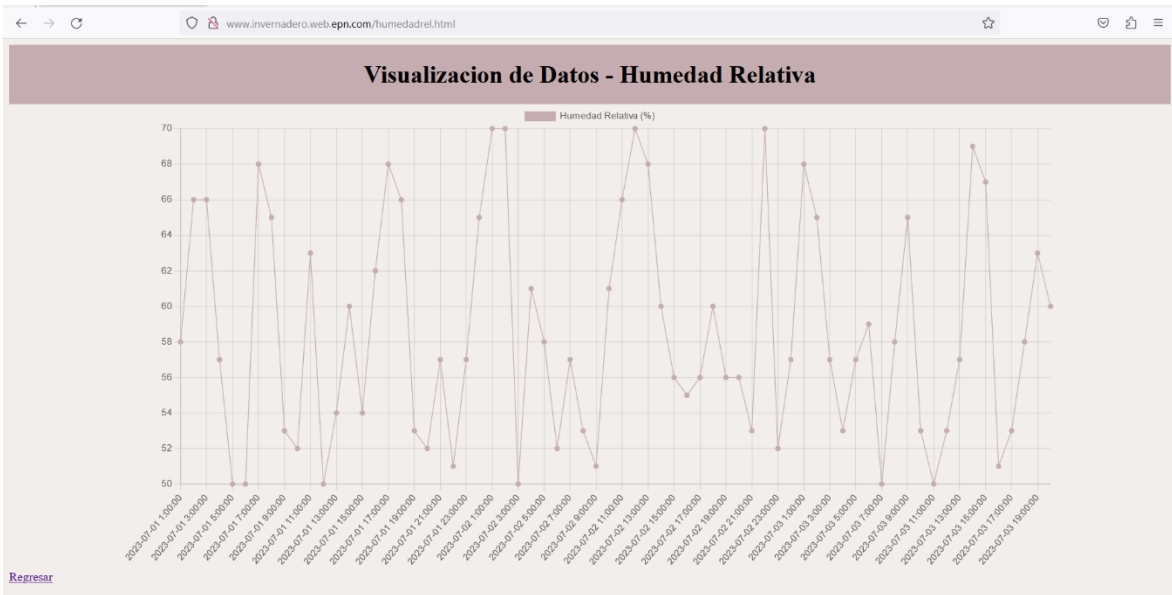
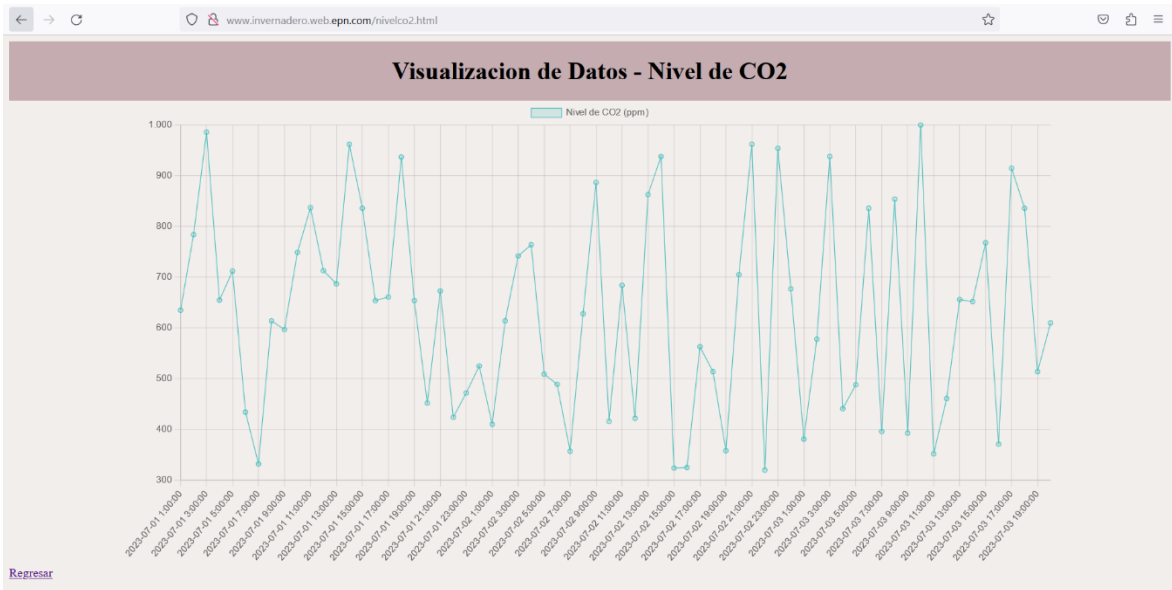


Figura 3.5. Gráfica de la variable humedad relativa.



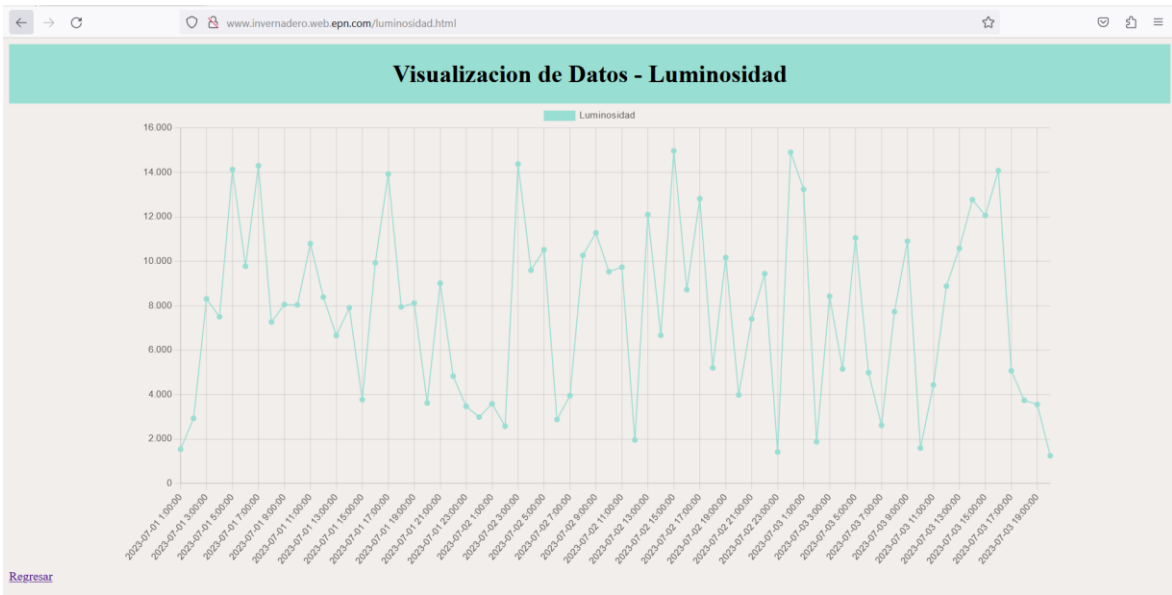


Figura 3.7. Gráfica de la variable luminosidad.

Por último, se tiene la visualización de todo el histórico de las variables que cuenta con la opción de descargar la tabla para manipular de mejor manera los valores capturados.

HISTORICO						
Fecha	Hora	Temperatura (°C)	Humedad (%)	Humedad Relativa (%)	Nivel de CO2 (ppm)	Luminosidad (luxes)
2023-07-01	1:00:00	23,00	40,00	58,00	635,00	1539,00
2023-07-01	2:00:00	17,00	58,00	66,00	784,00	2930,00
2023-07-01	3:00:00	17,00	33,00	66,00	986,00	8306,00
2023-07-01	4:00:00	25,00	55,00	57,00	655,00	7503,00
2023-07-01	5:00:00	26,00	61,00	50,00	712,00	14133,00
2023-07-01	6:00:00	25,00	76,00	50,00	434,00	9777,00
2023-07-01	7:00:00	16,00	38,00	68,00	332,00	14303,00
2023-07-01	8:00:00	19,00	58,00	65,00	614,00	7262,00
2023-07-01	9:00:00	26,00	58,00	53,00	597,00	8055,00
2023-07-01	10:00:00	25,00	54,00	52,00	749,00	8039,00
2023-07-01	11:00:00	24,00	72,00	63,00	837,00	10795,00
2023-07-01	12:00:00	15,00	79,00	50,00	713,00	8387,00
2023-07-01	13:00:00	18,00	35,00	54,00	687,00	6662,00
2023-07-01	14:00:00	21,00	74,00	60,00	962,00	7908,00
2023-07-01	15:00:00	19,00	68,00	54,00	836,00	3771,00
2023-07-01	16:00:00	20,00	41,00	62,00	654,00	9929,00
2023-07-01	17:00:00	17,00	66,00	68,00	661,00	13928,00

Figura 3.8. Histórico de valores de las variables.

3.2 Conclusiones

- Para la solución del presente trabajo, se tuvo que investigar acerca de los servicios que pueden ser usados por defecto en la placa Dragino Yum Shield para el diseño de una página web sin la necesidad de depender de servicios extras en la nube. Esto es posible debido a que la placa Dragino Yum Shield está diseñada en OpenWRT que es una distribución de Linux y tiene montado por defecto un servidor web denominado uHTTPd, y todos los archivos de diseño de la página pueden ser colocados directamente en el directorio /www/ debido a que es el directorio raíz para el servidor web que viene incluido en la placa.
- Una de las limitaciones que se tiene al configurar la placa Dragino Yum SHIELD, se debe a que si se desea usar el servidor en distintas redes Wi-Fi se debe de configurar nuevamente a la placa a la red que se requiera para que la placa adquiera una dirección. Este procedimiento se lo debe de realizar para cuando se requiera mover el servidor a otro lugar.
- Para tener la placa Dragino Yum Shield con sus últimos paquetes y librerías en el sistema operativo Linux, se puede actualizar el firmware de manera gráfica mediante WebUI o mediante líneas de comandos vía SSH, los softwares se pueden descargar de la página oficial de Dragino, pero se debe de tener en cuenta que algunas versiones no están activas por completo por lo cual puede ocasionar algunas limitantes en la placa.
- Al usar los servicios que ofrece la placa Dragino Yum Shield, se tiene algunas limitantes al momento de presentar la página web, una de estas es el tema de la base de datos de las variables a ser monitoreadas, debido que al tener una base de datos en uno de los servicios de la web como: MySQL, PostgreSQL, MongoDB ocupan capacidad de recursos de la placa Arduino Yum y el procesamiento de datos puede que sea un poco más deficiente. Es por esto por lo que se realizó la simulación de una base de datos de las variables a monitorear y visualizarla en la página web usando lenguaje de programación JavaScript junto con HTML y CSS.
- Se realizó la simulación de datos con un periodo de una hora debido a que el procesamiento de datos afectará directamente a la página web, es por eso se debe de realizar investigaciones previas para no saturar al sistema operativo, porque mientras mas datos en periodos mas pequeños de tiempo se guardan directamente en la memoria de la placa, por lo cual investigar acerca de base de datos en la nube o aplicaciones en la web puede solucionar esta limitante.

- El uso de la placa Dragino Yum Shield tuvo las mismas funcionalidades y beneficios que al usar la placa Yum la cual no estuvo en el mercado durante la ejecución de este proyecto, es por esto que haciendo uso del shield puede ser más accesible y mejor usado con otras placas permitiendo el uso de red Wi-Fi o internet cableado para realizar actividades relacionadas con IoT o invernaderos más completos.
- El monitoreo de datos tiene grandes ventajas, una de estas es económica debido a que solamente se necesita de una placa Dragino Yum Shield y un Arduino UNO que, si se implementa con todos los actuadores, sensores, motores, etc. se puede dar seguimiento sin hacer uso de productos costosos. Y sin mencionar de los servicios que pueden ser implementados en la nube que ofrece un seguimiento y monitoreo más robusto al invernadero.

3.3 Recomendaciones

- Al configurar la placa Arduino Shield al internet de distintas redes, es recomendable hacer uso de un programa que analice los puertos abiertos de los equipos que se están conectado en la red. El programa que se usó durante el desarrollo del documento fue Advanced Port Scanner, esto se debe a que una vez la placa esté configurada a una nueva red, la IP asignada al servidor web de Arduino se cambia a una dirección dentro del rango de dirección IP de la red. Esto facilita reconocer que IP se tiene asignada la placa y poder ingresar vía SSH para ser configurada.
- Se recomienda hacer uso de un editor de código para realizar las configuraciones pertinentes, debido a que si se desea configurar todo vía SSH con los editores vi, o nano, será muy complejo y no se podrá reconocer donde se tiene algún tipo de error de sintáxis dentro de las líneas de código. Para el presente documento se hizo uso del editor Visual Studio Code que permite crear de manera más sencilla los archivos en las distintas extensiones.
- Durante la creación del código, hubo la posibilidad de crear todas las gráficas de las variables dentro de un solo script, pero se recomienda que para reconocer algún error o editar de manera más sencilla la variable que se desea, se creó archivos independientes para cada una de las variables para tener el código mejor presentado y entendible para posibles cambios que puedan existir en un futuro.
- Se recomienda hacer uso del sketch que vienen de ejemplo para conocer cómo funciona las placas fusionadas, uno de ellos es hacer uso del sketch “blink” que su función es hacer parpadear un LED de la placa Arduino UNO y una vez familiarizado

con el funcionamiento se hace uso de la librería bridge para realizar programas más completos de acuerdo con las necesidades requeridas.

- El problema que se tiene al conectarse vía SSH es de seguridad, debido a que, si no se coloca protocolos de seguridad robustos, solamente teniendo la IP del servidor implementado se puede acceder y cambiar los parámetros y las variables pueden ser manipuladas.
- Se recomienda analizar previamente las capturas de datos y su periodo, debido a que una gran cantidad de datos puede saturar a la placa Dragino Yum Shield.

4 REFERENCIAS BIBLIOGRAFICAS

- [1] "Invernaderos: qué son y para qué tipo de cultivos se utilizan," Mundoriego. [En línea]. Disponible en: <https://mundoriego.es/invernaderos-que-son-y-para-que-tipo-de-cultivos-se-utilizan/>. [Accedido: 30 de junio de 2023].
- [2] "Análisis de riesgos. Trabajos en invernaderos," Universidad Complutense de Madrid. [En línea]. Disponible en: <https://www.ucm.es/data/cont/media/www/pag-56437/Analisis%20riesgos.%20trabajos%20en%20invernaderos.pdf>. [Accedido: 30 de junio de 2023].
- [3] "Arduino Yum Shield" Arduino. [En línea]. Disponible en: <https://docs.arduino.cc/retired/shields/genuino-yum-shield>. [Accedido: 30 de junio de 2023].
- [4] "Qué es un invernadero y un Fitotoldo" LIHARPERU. [En línea]. Disponible en: <https://lihar.com.pe/que-es-un-invernadero-y-un-fitotoldo/>. [Accedido: 30 de junio de 2023].
- [5] "Clima invernadero: cómo conseguir temperatura ideal," Novagric. [En línea]. Disponible en: <https://www.novagric.com/es/blog/articulos/clima-invernadero-como-conseguir-temperatura-ideal>. [Accedido: 30 de junio de 2023].
- [6] "Ideal Greenhouse Temperature and Humidity," Atlas Scientific. [En línea]. Disponible en: <https://atlas-scientific.com/blog/ideal-greenhouse-temperature-and-humidity/>. [Accedido: 1 de agosto de 2023].
- [7] "Ideal Greenhouse Temperature and Humidity," Atlas Scientific. [En línea]. Disponible en: <https://atlas-scientific.com/blog/ideal-greenhouse-temperature-and-humidity/>. [Accedido: 1 de agosto de 2023].
- [8] Sistemas Hortícolas Almería. "Humedad Relativa Ideal en Invernaderos." Sistemas Hortícolas Almería. Consultado en: agosto 2, 2023. [En línea]. Disponible en: <https://www.sistemashorticolasalmeria.com/blog/humedad-relativa-ideal/>
- [9] Agrofacto, "CO2 en el invernadero," [En línea]. Disponible: <https://agrofacto.com/co2-invernadero/>. [Último acceso: 2 de agosto de 2023].
- [10] "Cómo influye el dióxido de carbono (CO2) en el cultivo en invernadero," Nutricontrol. [En línea]. Disponible en: <https://nutricontrol.com/es/como-influye-el-dioxido-de-carbono-co2-en-el-cultivo-en-invernadero/>. [Accedido: 6 de agosto de 2023]

- [11] Hydroenv, "Catálogo de Productos," [En línea]. Disponible en: https://www.hydroenv.com.mx/catalogo/index.php?main_page=page&id=221. [Accedido: 6 de agosto de 2023].
- [12] P. Melgareto, A. Navarro, P. Legua y V. Lidón, "La Iluminación en los Invernaderos," [En línea]. Disponible en: <http://dspace.umh.es/bitstream/11000/5465/1/La%20Iluminaci%C3%B3n%20en%20los%20Invernaderos.pdf>. [Último acceso: 6 de agosto de 2023].
- [13] "Introduction," Arduino. [En línea]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>. [Accedido: 9 de agosto de 2023]
- [14] "Arduino Uno Rev3 Board," Arduino Documentation. [En línea]. Disponible en: <https://docs.arduino.cc/hardware/uno-rev3>. [Accedido: 9 de agosto de 2023].
- [15] "¿Qué es un Shield?," Arduino.cl. [En línea]. Disponible en: <https://arduino.cl/que-es-un-shield/>. [Accedido: 9 de agosto de 2023].
- [16] "Genuino Yum Shield," Arduino Documentation. [En línea]. Disponible en: <https://docs.arduino.cc/retired/shields/genuino-yum-shield>. [Accedido: 9 de agosto de 2023].
- [17] "Dragino Yum Shield V2.4 Linux WiFi Ethernet USB All-in-One Compatible with Arduino Leonardo Uno Duemilanove Diecimila Mega Teensy," Desertcart.ec. [En línea]. Disponible en: <https://www.desertcart.ec/products/31155805-dragino-yum-shield-v2-4-linux-wifi-ethernet-usb-all-in-one-compatible-with-arduino-leonardo-uno-duemilanove-diecimila-mega-teensy>. [Accedido: 9 de agosto de 2023].
- [18] "Yum Shield," Altronics. [En línea]. Disponible en: <https://altronics.cl/yum-shield>. [Accedido: 18 de agosto de 2023].
- [19] "¿Qué es un servidor y cómo funciona?," Rock Content. [En línea]. Disponible en: <https://rockcontent.com/es/blog/que-es-un-servidor/>. [Accedido: 18 de agosto de 2023].
- [20] "Arduino Yum," Arduino Documentation. [En línea]. Disponible en: <https://docs.arduino.cc/retired/getting-started-guides/ArduinoYum#web-services>. [Accedido: 18 de agosto de 2023].
- [21] "Arduino Yum - Intro to Web Server," Arduino Project Hub. [En línea]. Disponible en: https://projecthub.arduino.cc/Arduino_Scuola/arduino-yum-intro-to-web-server-49bcf3. [Accedido: 18 de agosto de 2023].

- [22] "HTML Básico," Mozilla Developer Network. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics. [Accedido: 18 de agosto de 2023].
- [23]"CSS Básico," Mozilla Developer Network. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics. [Accedido: 18 de agosto de 2023].
- [24]JavaScript Básico," Mozilla Developer Network. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics. [Accedido: 18 de agosto de 2023].
- [25] MobaXterm, "Enhanced terminal with an X server and a set of Unix commands (GNU/Cygwin) - Mobatek," Fuente en línea. Disponible en: <https://mobaxterm.mobatek.net/>. [Accedido: 21 de agosto de 2023].
- [26] IEBSchool, "¿Qué es una etiqueta HTML? Analítica y usabilidad," IEBSchool Blog, 12-ago-2021. [En línea]. Disponible en: <https://www.iebschool.com/blog/que-es-etiqueta-html-analitica-usabilidad/> [Accedido: 21 de agosto de 2023].
- [27] Caronte Studio, "Listado de Etiquetas HTML," Caronte Studio Blog, 10-jul-2022. [En línea]. Disponible en: <https://carontestudio.com/blog/listado-de-etiquetas-html/> [Accedido: 21 de agosto de 2023]
- [28] Eniun. "Resumen de Tabla de Propiedades CSS y Valores." Eniun. Disponible en: <https://www.eniun.com/resumen-tabla-propiedades-css-valores/>. [Accedido: 21 de agosto de 2023].
- [29] Adictos al Trabajo. (2022, Julio 1). "Cómo añadir gráficos en tu web con Chart.js." Adictos al Trabajo. [En línea]. Disponible en: <https://www.adictosaltrabajo.com/2022/07/01/como-anadir-graficos-en-tu-web-con-chart-js/>. [Accedido: 23 de agosto de 2023].
- [30] Kio4.com, "100 Proyectos de Arduino." Kio4.com. [En línea]. Disponible en: http://kio4.com/arduino/100arduino_yum.htm. [Accedido: 23 de agosto de 2023].
- [31] "Bridge Library - Arduino Reference." [En línea]. Disponible en: <https://www.arduino.cc/reference/en/libraries/bridge/>. [Accedido: 23 de agosto de 2023].
- [32] Panamahitek. (2017, Julio 27). "Instalación de PHP en Arduino Yum." Panamahitek. [En línea]. Disponible en: <https://panamahitek.com/instalacion-de-php-arduino-yum/>. [Accedido: 23 de agosto de 2023].

[33] "uHTTPd - OpenWrt Wiki." [En línea]. Disponible en: <https://openwrt.org/docs/guide-user/services/webserver/uhttpd>. [Consultado: Agosto 23, 2023].

[34] "YumBridgeLibrary - Arduino Reference." [En línea]. Disponible en: <https://docs.arduino.cc/retired/archived-libraries/YumBridgeLibrary>. [Consultado: Agosto 23, 2023].

5 ANEXOS

ANEXO I. Código fuente del archivo index.html.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>INVERNADERO EPN</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body class="portada">
  <h1 class="t1">ESCUELA POLITECNICA NACIONAL <br>
  FACULTAD DE INGENIERIA ELÉCTRICA Y ELECTRÓNICA <br>
  TRABAJO DE INTEGRACIÓN CURRICULAR <br> </h1>
  <p></p></p>
  <h2 class="t2"> SISTEMA DE RIEGO PARA INVERNADEROS CON INTERFAZ
<br>
  WEB UTILIZANDO UN ARDUINO YUM </h2>
  <div>
    <h2 class="t3"> Estudiante: Xavier Chasi<br>
    Director: Eduardo Morejon </h2>
  </div>
  <div>
    <a class="boton" href="datos.html"><b>INGRESAR</b> </a>
  </div>
</body>
</html>
```

ANEXO II. Código fuente del archivo datos.html.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <script src="script.js"></script>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Página de Datos</title>
</style>
</head>
<body class="inver">
  <div class="titulo">
    <h1>MONITOREO DEL INVERNADERO</h1>
  </div>

  <div class="contenedor-secciones">
    <!-- GRAFICA TEMPERATURA -->
    <div class="seccion temperatura">
      <a class="botontem" href="temperatura.html"><div
class="item">&#x1f321;&#xfe0f</div></a>
      <h1>Temperatura</h1>
      <!-- MAX -->
      <div>
        <p>Valor máximo de temperatura: <span id="maxTemp"></span> </p>
        <p>Valor mínimo de temperatura: <span id="minTemp"></span> </p>
        <p>Promedio de temperatura: <span id="averageTemp"></span> </p>
      </div>
    </div>

    <!-- GRAFICA HUMEDAD -->
    <div class="seccion humedad">
```

```

    <a class=botonhum href="humedad.html"><div
class="item">&#x1f331</div></a>

    <h1>Humedad</h1>

    <div>

        <p>Valor máximo de humedad: <span id="maxHume"></span> %</p>
        <p>Valor mínimo de humedad: <span id="minHume"></span> %</p>
        <p>Promedio de humedad: <span id="averageHume"></span> %</p>
    </div>
</div>

<!-- GRAFICA HUMEDAD RELATIVA -->
<div class="seccion relativa">

    <a class=botonhumr href="humedadrel.html"><div
class="item">&#128027</div></a>

    <h1>Humedad Relativa</h1>

    <div>

        <p>Valor máximo de humedad relativa: <span
id="maxHumeRel"></span> </p>

        <p>Valor mínimo de humedad relativa: <span
id="minHumeRel"></span> </p>

        <p>Promedio de humedad relativa: <span
id="averageHumeRel"></span> </p>
    </div>
</div>

<!-- GRAFICA NIVEL DE CO2 -->
<div class="seccion nivelc">

    <a class=botonco2 href="nivelco2.html"><div
class="item">&#x1f4a6</div></a>

    <h1>Nivel de Co2</h1>

    <div>

        <p>Valor máximo del nivel de CO2: <span id="maxCO2"></span> </p>
        <p>Valor mínimo del nivel de CO2: <span id="minCO2"></span> </p>
        <p>Promedio del nivel de CO2: <span id="averageCO2"></span> </p>
    </div>
</div>

```



```

<!-- GRAFICA LUMINOSIDAD -->
<div class="seccion luminosidad">
  <a class=botonlum href="luminosidad.html"><div
class="item">&#x1f324;&#xfe0f</div></a>
  <h1>Luminosidad</h1>
  <div>
    <p>Valor máximo de humedad relativa: <span
id="maxLumino"></span> Luxes </p>
    <p>Valor mínimo de humedad relativa: <span
id="minLumino"></span> Luxes </p>
    <p>Promedio de humedad relativa: <span
id="averageLumino"></span> Luxes </p>
  </div>
</div>

<!-- TABLA HISTORICO -->
<div class="seccion historico">
  <a class=botonhis href="historico.html"><div
class="item">&#x1f4cb</div></a>
  <h1>Historico</h1>
</div>

  <div>
    <a class="back" href="index.html">Inicio</a>
  </div>

</body>
</html>

```

ANEXO III. Código fuente del archivo temperatura.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Visualización de Datos - Temperatura</title>
  <style>
    body{
      background-color: #F2EEEE;
    }
    .chart-container {
      overflow-x: auto; /* Habilita la barra de desplazamiento
horizontal */
      width: 80%; /* Ancho del contenedor */
      margin: 0 auto; /* Centra el contenedor horizontalmente */
    }

    .titulo h1{
      text-align: center;
      background-color: rgba(255, 99, 132, 0.2);
      padding: 20px;
      margin: 0;
    }
  </style>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <script src="temperatura_grafica.js"></script>
</head>
<body>
  <div class="titulo">
    <h1>Visualización de Datos - Temperatura</h1>
  </div>

  <div class="chart-container">
```

```

        <canvas id="temperatureChart"></canvas>
    </div>
    <div>
        <a class="bt" href="datos.html">Regresar</a>
    </div>
</body>
</html>

```

ANEXO IV. Código fuente del archivo humedad.html

```

<!DOCTYPE html>
<html>
<head>
    <title>Visualización de Datos - Humedad</title>
    <style>
        body{
            background-color: #F2EEEE;
        }

        .titulo h1{
            text-align: center;
            background-color: rgba(255, 99, 132, 0.2);
            padding: 20px;
            margin: 0;
        }

        .chart-container {
            overflow-x: auto; /* Habilita la barra de desplazamiento
horizontal */
            width: 80%; /* Ancho del contenedor */
            margin: 0 auto; /* Centra el contenedor horizontalmente */
        }
    </style>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script src="humedad_grafica.js"></script>

```

```

</head>
<body>
  <div class="titulo">
    <h1>Visualización de Datos - Humedad</h1>
  </div>

  <div class="chart-container">
    <canvas id="humidityChart"></canvas>
  </div>

  <div>
    <a class="bt" href="datos.html">Regresar</a>
  </div>
</body>
</html>

```

ANEXO V. Código fuente del archivo humedadrel.html

```

<!DOCTYPE html>
<html>
<head>
  <title>Visualización de Datos - Humedad Relativa</title>
  <style>
    body{
      background-color: #F2EEEE;
    }
    .titulo h1{
      text-align: center;
      background-color: rgb(196, 172, 176);
      padding: 20px;
      margin: 0;
    }
    .chart-container {
      overflow-x: auto; /* Habilita la barra de desplazamiento
horizontal */

```

```

        width: 80%; /* Ancho del contenedor */
        margin: 0 auto; /* Centra el contenedor horizontalmente */
    }
</style>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script src="grafica_humedadrel.js"></script>
</head>
<body>
    <div class="titulo">
        <h1>Visualización de Datos - Humedad Relativa</h1>
    </div>

    <div class="chart-container">
        <canvas id="humidityRelChart"></canvas>
    </div>

    <div>
        <a class="bt" href="datos.html">Regresar</a>
    </div>
</body>
</html>

```

ANEXO VI. Código fuente del archivo nivelco2.html

```

<!DOCTYPE html>
<html>
<head>
    <title>Visualización de Datos - Nivel de CO2</title>
    <style>

        body{
            background-color: #F2EEEE;
        }

        .chart-container {

```

```

        overflow-x: auto; /* Habilita la barra de desplazamiento
horizontal */
        width: 80%; /* Ancho del contenedor */
        margin: 0 auto; /* Centra el contenedor horizontalmente */
    }
    .titulo h1{
        text-align: center;
        background-color: rgb(196, 172, 176);
        padding: 20px;
        margin: 0;
    }

</style>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script src="grafica_co2.js"></script>
</head>
<body>
    <div class="titulo">
        <h1>Visualización de Datos - Nivel de CO2</h1>
    </div>
    <div class="chart-container">
        <canvas id="co2Chart"></canvas>
    </div>
    <div>
        <a class="bt" href="datos.html">Regresar</a>
    </div>
</body>
</html>

```

ANEXO VII. Código fuente del archivo luminosidad.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Visualización de Datos - Luminosidad</title>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <script src="grafica_luminosidad.js"></script>
  <style>
    body{
      background-color: #F2EEEE;
    }
    .chart-container {
      overflow-x: auto; /* Habilita la barra de desplazamiento
horizontal */
      width: 80%; /* Ancho del contenedor */
      margin: 0 auto; /* Centra el contenedor horizontalmente */
    }
    .titulo h1{
      text-align: center;
      background-color: #98DED3;
      padding: 20px;
      margin: 0;
    }
  </style>
</head>
<body>
  <div class="titulo">
    <h1>Visualización de Datos - Luminosidad</h1>
  </div>

  <div class="chart-container">
    <canvas id="luminosityChart"></canvas>
  </div>
</body>
</html>
```

```
<div>
  <a class="bt" href="datos.html">Regresar</a>
</div>
</body>
</html>
```

ANEXO VIII. Código fuente del archivo historico.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Historial Completo</title>
  <style>
    /* Estilos para la tabla */
    table {
      border-collapse: collapse;
      width: 100%;
      font-family: Arial, sans-serif;
    }

    th, td {
      border: 1px solid #dddddd;
      text-align: center;
      padding: 8px;
    }

    th {
      background-color: #f2f2f2;
    }

    .titulo h1{
      text-align: center;
      background-color: rgb(196, 172, 176);
```



```

        padding: 20px;
        margin: 0;
    }

    tr:first-child th {
        background-color: rgb(196, 172, 176); /* Cambia el color de fondo a tu
preferencia */
    }
</style>
</head>
<body>
    <div class="titulo">
        <h1>HISTORICO</h1>
    </div>
    <button id="downloadButton">Descargar CSV</button>
    <table>
        <tr>
            <th>Fecha</th>
            <th>Hora</th>
            <th>Temperatura (°C)</th>
            <th>Humedad</th>
            <th>Humedad Relativa</th>
            <th>Nivel de CO2</th>
            <th>Luminosidad</th>
        </tr>
        <!-- Aquí se llenarán los datos desde el CSV usando JavaScript -->
    </table>
    <a class="bt" href="datos.html">Regresar</a>

    <script>
        document.getElementById("downloadButton").addEventListener("click"
, function() {
            var table = document.querySelector("table");
            var csv = [];

```

```

var rows = table.querySelectorAll("tr");
for (var i = 0; i < rows.length; i++) {
    var row = [], cols = rows[i].querySelectorAll("td, th");
    for (var j = 0; j < cols.length; j++) {
        row.push(cols[j].innerText);
    }
    csv.push(row.join(","));
}
var csvText = csv.join("\n");
var blob = new Blob([csvText], { type: "text/csv;charset=utf-8;" });
var link = document.createElement("a");
var url = URL.createObjectURL(blob);
link.href = url;
link.download = "tabla.csv";
link.click();
URL.revokeObjectURL(url);
});

```

```
</script>
```

```
<script>
```

```

fetch("basedatos.csv")
    .then(response => response.text())
    .then(data => {
        var lines = data.split('\n');
        var tableHTML = "";

        for (var i = 1; i < lines.length; i++) {
            var parts = lines[i].split(';');

            // Verificar si la línea tiene suficientes partes (campos)
            if (parts.length >= 7) {
                tableHTML += "<tr>";
            }
        }
    });

```

```
        tableHTML += "<td>" + parts[0] + "</td>";
        tableHTML += "<td>" + parts[1] + "</td>";
        tableHTML += "<td>" + parts[2] + "</td>";
        tableHTML += "<td>" + parts[3] + "</td>";
        tableHTML += "<td>" + parts[4] + "</td>";
        tableHTML += "<td>" + parts[5] + "</td>";
        tableHTML += "<td>" + parts[6] + "</td>";
        tableHTML += "</tr>";
    }
}

document.querySelector("table").innerHTML += tableHTML;
});

</script>
</body>
</html>
```

ANEXO IX. Código fuente del archivo styleless.css

```
body{
    text-align: center;
}
.portada {
    background-image: url("foto.webp");
    background-repeat: no-repeat;
    background-size: cover;
}
.titulo{
    font-size: large;
}
```

```

.t1{

    background-color: #F2EEEE;
    color: black;
    display: inline-block;
}

.t2{

    background-color: #F2EEEE;
    color: black;
    display: inline-block;
}

.t3{

    background-color: #F2EEEE;
    color: black;
    display: inline-block;
    text-align: justify;
}

.imagen{

    background: #F2EEEE; /* Fondo negro semitransparente */
    display: inline-block; /* Asegurar que el contenedor se ajuste al
contenido */
    padding: 10px;
    width: 20%;
    height: 60%;
}

.boton {

    text-decoration: none;
    background-color: black;
    color: white;
    display: inline-block;
}

```

```
border-radius: 15px;
padding: 15px;
}

.titulo {
padding: 5px;
text-align: center;
margin-top: 20px;
font-size: 24px;
border-radius: 10px;
background-color: #799CB0;
color: white;
}

.inver{
background-color: #F2EEEE;
}

.seccion {
padding: 20px;
text-align: center;
margin: 20px;
border-radius: 10px;
color: white;
font-size: 18px;
width: 400px; /* Ancho máximo de la sección */
}

.contenedor-secciones {
display: flex;
flex-direction: column;
```

```
    align-items: center;
}

.temperatura {
    background-color: #ff5733; /* Color para temperatura */
    text-align: center;
}

.Historial{
    text-align: center;
}

.item{
    font-size: 70px;
}

.humedad {
    background-color: #DD5A7A; /* Color para humedad */
}

.relativa {
    background-color: #ECCFD3; /* Color para humedad */
}

.nivelc {
    background-color: #F0E2C5; /* Color para humedad */
}

.luminosidad {
    background-color: #98DED3; /* Color para humedad */
}

.historico {
    background-color: #FFCEB7; /* Color para humedad */
}
```

```
}  
  
.back{  
    text-decoration: none;  
    background-color: #615887;  
    color: white;  
    display: inline-block;  
    border-radius: 10px;  
    padding: 10px;  
}  
  
.pagtem{  
    text-decoration: none;  
    background-color: #ff5733;  
    color: white;  
    display: inline-block;  
    border-radius: 10px;  
    padding: 10px;  
}  
  
.botontem{  
    text-decoration: none;  
    background-color: #ff5733;  
    color: white;  
    display: inline-block;  
    border-radius: 10px;  
    padding: 10px;  
}  
  
.botonhum{  
    text-decoration: none;  
    color: white;  
    display: inline-block;  
    border-radius: 10px;  
    padding: 10px;  
}
```

```
}  
.botonhumr{  
    text-decoration: none;  
    background-color: #ECCFD3;  
    color: white;  
    display: inline-block;  
    border-radius: 10px;  
    padding: 10px;  
}  
.botonco2{  
    text-decoration: none;  
    background-color: #F0E2C5;  
    color: white;  
    display: inline-block;  
    border-radius: 10px;  
    padding: 10px;  
}  
.botonlum{  
    text-decoration: none;  
    background-color: #98DED3;  
    color: white;  
    display: inline-block;  
    border-radius: 10px;  
    padding: 10px;  
}  
.botonhis{  
    text-decoration: none;  
    background-color: #FFCEB7;  
    color: white;  
    display: inline-block;  
    border-radius: 10px;  
    padding: 10px;  
}
```


ANEXO X. Código fuente del archivo temperatura_grafica.js

```
window.onload = function () {
  // Cargar el archivo CSV y procesar los datos
  fetch("basedatos.csv")
    .then(response => response.text())
    .then(data => {
      var lines = data.split('\n');
      var labels = [];
      var temperatureData = [];
      var humidityData = [];

      for (var i = 6; i < lines.length - 1; i++) {
        var parts = lines[i].split(';');
        var fechaHora = parts[0] + " " + parts[1];
        var temperatura = parseFloat(parts[2]); // Columna C
        (Temperatura)
        var humedad = parseFloat(parts[3]);

        labels.push(fechaHora);
        humidityData.push(humedad);
        temperatureData.push(temperatura);
      }

      // GRAFICAR

      const temperatureCtx =
document.getElementById('temperatureChart').getContext('2d');
      new Chart(temperatureCtx, {
        type: 'line',
        data: {
          labels: labels,
          datasets: [{
            label: 'Temperatura (°C)',
```

```

        data: temperatureData,
        borderColor: 'rgba(255, 99, 132, 1)',
        backgroundColor: 'rgba(255, 99, 132, 0.2)',
        borderWidth: 1
    }]
},
options: {
    scales: {
        y: {
            beginAtZero: false
        }
    }
}
});
});
}

```

ANEXO XI. Código fuente del archivo humedad_grafica.js

```

window.onload = function () {
    fetch("basedatos.csv")
        .then(response => response.text())
        .then(data => {
            var lines = data.split('\n');
            var labels = [];
            var humidityData = [];

            for (var i = 6; i < lines.length-1; i++) {
                var parts = lines[i].split(';');
                var fechaHora = parts[0] + " " + parts[1];
                var humedad = parseFloat(parts[3]); // Cambia el índice
                para la variable de humedad

                labels.push(fechaHora);
            }
        })
}

```

```

        humidityData.push(humedad);
    }

    const humidityCtx =
document.getElementById('humidityChart').getContext('2d');
    new Chart(humidityCtx, {
        type: 'line',
        data: {
            labels: labels,
            datasets: [{
                label: 'Humedad (%)',
                data: humidityData,
                borderColor: 'rgb(221, 90, 122) ',
                backgroundColor: 'rgb(221, 90, 122)',
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: false
                }
            }
        }
    });
});
}

```

ANEXO XII. Código fuente del archivo grafica_humedadrel.js

```
window.onload = function () {
  fetch("basedatos.csv")
    .then(response => response.text())
    .then(data => {
      var lines = data.split('\n');
      var labels = [];
      var humidityRelData = [];

      for (var i = 6; i < lines.length - 1; i++) {
        var parts = lines[i].split(';');
        var fechaHora = parts[0] + " " + parts[1];
        var humedadRelativa = parseFloat(parts[4]); // Columna E
        (Humedad Relativa)

        labels.push(fechaHora);
        humidityRelData.push(humedadRelativa);
      }

      const humidityRelCtx =
document.getElementById('humidityRelChart').getContext('2d');
      new Chart(humidityRelCtx, {
        type: 'line',
        data: {
          labels: labels,
          datasets: [{
            label: 'Humedad Relativa (%)',
            data: humidityRelData,
            borderColor: 'rgb(196, 172, 176)',
            backgroundColor: 'rgb(196, 172, 176)',
            borderWidth: 1
          }]
        },
      },
```

```

        options: {
            scales: {
                y: {
                    beginAtZero: false
                }
            }
        }
    });
});
}

```

ANEXO XIII. Código fuente del archivo grafica_co2.js

```

window.onload = function () {
    fetch("basedatos.csv")
        .then(response => response.text())
        .then(data => {
            var lines = data.split('\n');
            var labels = [];
            var co2Data = [];

            for (var i = 6; i < lines.length - 1; i++) {
                var parts = lines[i].split(';');
                var fechaHora = parts[0] + " " + parts[1];
                var nivelCO2 = parseFloat(parts[5]); // Columna F (Nivel
de CO2)

                labels.push(fechaHora);
                co2Data.push(nivelCO2);
            }

            const co2Ctx =
document.getElementById('co2Chart').getContext('2d');
            new Chart(co2Ctx, {

```

```

    type: 'line',
    data: {
      labels: labels,
      datasets: [{
        label: 'Nivel de CO2 (ppm)',
        data: co2Data,
        borderColor: 'rgba(75, 192, 192, 1)',
        backgroundColor: 'rgba(75, 192, 192, 0.2)',
        borderWidth: 1
      }]
    },
    options: {
      scales: {
        y: {
          beginAtZero: false
        }
      }
    }
  });
});
}

```

ANEXO XIV. Código fuente del archivo grafica_luminosidad.js

```

<!DOCTYPE html>
<html>
<head>
  <title>Visualización de Datos - Luminosidad</title>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <script src="grafica_luminosidad.js"></script>
  <style>
    body{
      background-color: #F2EEEE;
    }
  </style>

```

```

    }
    .chart-container {
        overflow-x: auto; /* Habilita la barra de desplazamiento
horizontal */
        width: 80%; /* Ancho del contenedor */
        margin: 0 auto; /* Centra el contenedor horizontalmente */
    }
    .titulo h1{
        text-align: center;
        background-color: #98DED3;
        padding: 20px;
        margin: 0;
    }
</style>
</head>
<body>
    <div class="titulo">
        <h1>Visualización de Datos - Luminosidad</h1>
    </div>

    <div class="chart-container">
        <canvas id="luminosityChart"></canvas>
    </div>

    <div>
        <a class="bt" href="datos.html">Regresar</a>
    </div>
</body>
</html>

```

ANEXO XV. Código fuente del archivo script.js

```
window.onload = function () {  
    // Cargar el archivo CSV y procesar los datos  
    fetch("basedatos.csv")  
        .then(response => response.text())  
        .then(data => {  
            var lines = data.split('\n');  
            var tempe = []; // Arreglo para almacenar las temperaturas  
            var hume = []; // Arreglo para almacenar las humedades  
            var humeRel = []; // Arreglo para almacenar las humedades  
relativas  
            var co2 = []; // Arreglo para almacenar los niveles de CO2  
            var lumino = []; // Arreglo para almacenar las luminosidades  
  
            for (var i = 6; i < lines.length - 1; i++) {  
                var parts = lines[i].split(';');  
                var temperatura = parseFloat(parts[2]); // Columna C  
(Temperatura)  
                var humedad = parseFloat(parts[3]); // Columna D (Humedad)  
                var humedadRelativa = parseFloat(parts[4]); // Columna E  
(Humedad Relativa)  
                var nivelCO2 = parseFloat(parts[5]); // Columna F (Nivel  
de CO2)  
                var luminosidad = parseFloat(parts[6]); // Columna G  
(Luminosidad)  
  
                tempe.push(temperatura);  
                hume.push(humedad);  
                humeRel.push(humedadRelativa);  
                co2.push(nivelCO2);  
                lumino.push(luminosidad);  
            }  
  
            // Calcular valores estadísticos  
            var maxTemp = Math.max(...tempe);
```



```

        var minTemp= Math.min(...tempe);
        var sumTemp = tempe.reduce((total, current) => total +
current, 0);
        var averageTemp = sumTemp / tempe.length;

        var maxHume = Math.max(...hume);
        var minHume= Math.min(...hume);
        var sumHume = hume.reduce((total, current) => total + current,
0);
        var averageHume = sumHume / hume.length;

        var maxHumeRel = Math.max(...humeRel);
        var minHumeRel = Math.min(...humeRel);
        var sumHumeRel = humeRel.reduce((total, current) => total +
current, 0);
        var averageHumeRel = sumHumeRel / humeRel.length;

        var maxCO2 = Math.max(...co2);
        var minCO2 = Math.min(...co2);
        var sumCO2 = co2.reduce((total, current) => total + current,
0);
        var averageCO2 = sumCO2 / co2.length;

        var maxLumino = Math.max(...lumino);
        var minLumino = Math.min(...lumino);
        var sumLumino = lumino.reduce((total, current) => total +
current, 0);
        var averageLumino = sumLumino / lumino.length;

        // Actualizar el contenido de los elementos HTML con los
valores estadísticos
        document.getElementById("maxTemp").textContent =
maxTemp.toFixed(2) + " °C";
        document.getElementById("minTemp").textContent =
minTemp.toFixed(2) + " °C";
        document.getElementById("averageTemp").textContent =
averageTemp.toFixed(2) + " °C";

```

```

        document.getElementById("maxHume").textContent =
maxHume.toFixed(2);

        document.getElementById("minHume").textContent =
minHume.toFixed(2);

        document.getElementById("averageHume").textContent =
averageHume.toFixed(2);

        document.getElementById("maxHumeRel").textContent =
maxHumeRel.toFixed(2) + " %";

        document.getElementById("minHumeRel").textContent =
minHumeRel.toFixed(2) + " %";

        document.getElementById("averageHumeRel").textContent =
averageHumeRel.toFixed(2) + " %";

        document.getElementById("maxCO2").textContent =
maxCO2.toFixed(2) + " ppm";

        document.getElementById("minCO2").textContent =
minCO2.toFixed(2) + " ppm";

        document.getElementById("averageCO2").textContent =
averageCO2.toFixed(2) + " ppm";

        document.getElementById("maxLumino").textContent =
maxLumino.toFixed(2);

        document.getElementById("minLumino").textContent =
minLumino.toFixed(2);

        document.getElementById("averageLumino").textContent =
averageLumino.toFixed(2);

        // Otro
    });
}

```

ANEXO XVI. Código fuente IDE Arduino.

```
#include <Bridge.h>
#include <Process.h> //Libreria de procesos para usar comandos
Linux en
#include <Console.h>
void setup() {
  Bridge.begin();
  Console.begin();
  Console.println("\n");
  Console.println("Invernadero EPN");
  Console.println("\nDatos del Sistema de Invernadero enviados
 exitosadamente!");
  randomSeed(analogRead(0)); // Inicializar la generación de
 números aleatorios
}

void loop() {
  //VARIABLES DEL INVERNADERO - VALORES ALEATORIOS
  String dataString = "content=Variables&";
  dataString += "temperatura=";
  dataString += String(random(14) + 28);
  dataString += "&";
  dataString += "humedad=";
  dataString += String(random(31) + 51);
  dataString += "&";
  dataString += "humedadRelativa=";
  dataString += String(random(21) + 40);
  dataString += "&";
  dataString += "NivelCO2=";
  dataString += String(random(401) + 800);
  dataString += "&";
  dataString += "luminosidad=";
  dataString += String(random(40001) + 10000);
  Console.println(dataString);
  Process p;
  p.begin("curl");
  p.addParameter("-k");
  p.addParameter("-X");
  p.addParameter("POST");
  p.addParameter("-H");
  p.addParameter("Content-Type: application/x-www-form-
 urlencoded");
  p.addParameter("-d");
  p.addParameter(dataString);
  p.addParameter("http://192.168.100.107/data_receiver.php"); // IP
del servidor
  p.run();

  while (p.running()) {
    // Esperar a que el proceso termine
  }
  delay(5000); // Esperar 5 segundos antes de enviar la siguiente
solicitud
}
```