# ESTACIÓN METEOROLÓGICA CON ESP32 Y BLYNK

Imagen que contiene azul, tabla, oscuro, escritorio

Descripción generada automáticamente

## ÍNDICE

## RESUMEN

El ESP32 es un microcontrolador muy potente el cual tiene la posibilidad de conectarse a la red WIFI, por medio de ellos podemos realizar lecturas de sensores y monitoreos por medio de IOT.

La estación meteorológica puede medir la temperatura y humedad del ambiente, también posee un anemómetro con el cual se puede medir la velocidad del viento, funciona de forma remota y se alimenta mediante baterías, los datos son enviados por medio de internet a un servicio en la nube de IoT llamado Blynk desde el cual se pueden monitorear los datos que el Esp32 detecta, así como ver su comportamiento a través del tiempo.

## COMPONENTES Y CONEXIONES

### Equipo y software necesarios:

Para realizar este proyecto será necesario poseer los siguientes elementos:

* Un módulo ESP32 y su cable de conexión.
* Una computadora que posea Arduino IDE (Preferiblemente con una versión actualizada del programa).
* Las librerías y controladores del ESP32 que se muestran como instalarse en la guía 1.
* Las librerías de Blynk las cuales pueden instalarse desde la página web o siguiendo los pasos de la guía 3.
* Modulo DHT22.
* Modulo de sensor de barrera infrarrojo.
* Cables para realizar conexiones.
* Regulador de voltaje LM7805 u otro dispositivo que permite reducir el voltaje.
* Baterías de litio y porta baterías.
* Partes impresas o estructura equivalente (Archivos STL en recursos).

### Temperatura y humedad

La temperatura y la humedad son medidos mediante el módulo DHT22 el cual posee un sensor AM2302, funciona desde 3,3 a 5 voltios puede medir temperaturas desde -40 °C hasta 80° C y posee una precisión para medir la humedad en el ambiente del 99.9%, posee tres pines los cuales corresponden a GND, VCC y salida de datos.

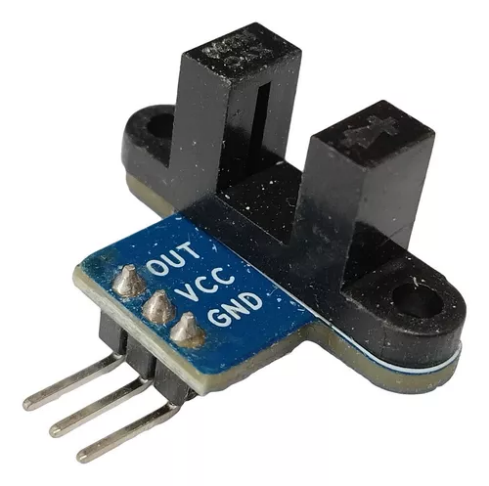


***Figura 1: Sensor de Temperatura y humedad DHT22***

El sensor esta conectado a la alimentación del ESP32 en los pines de GND para conexión a tierra y en VIN para la conexión a 5 voltios, el pin de salida esta conectado al pin número 13 de ESP32 el cual está declarada como una entrada.

### Velocidad del viento

El anemómetro el cual es un sistema encargado de la medición del viento está compuesto mediante tres aspas cóncavas las cuales hacen girar un eje que está conectado a un disco dentado, este disco posee 60 dientes, estos dientes pasan por un sensor de barrera de luz el cual esta compuesto por un led infrarrojo y un fotodetector, al pasar cada diente por el sensor obstaculiza el paso de la luz haciendo que el sensor informe un estado lógico al ESP32, al recibir este estado se aumenta un contador con el cual se realizan cálculos en base al diámetro del disco y la cantidad de dientes detectados y así tener una lectura aproximada de la velocidad del viento.

******

***Figura 2: Sensor de barrera infrarrojo***

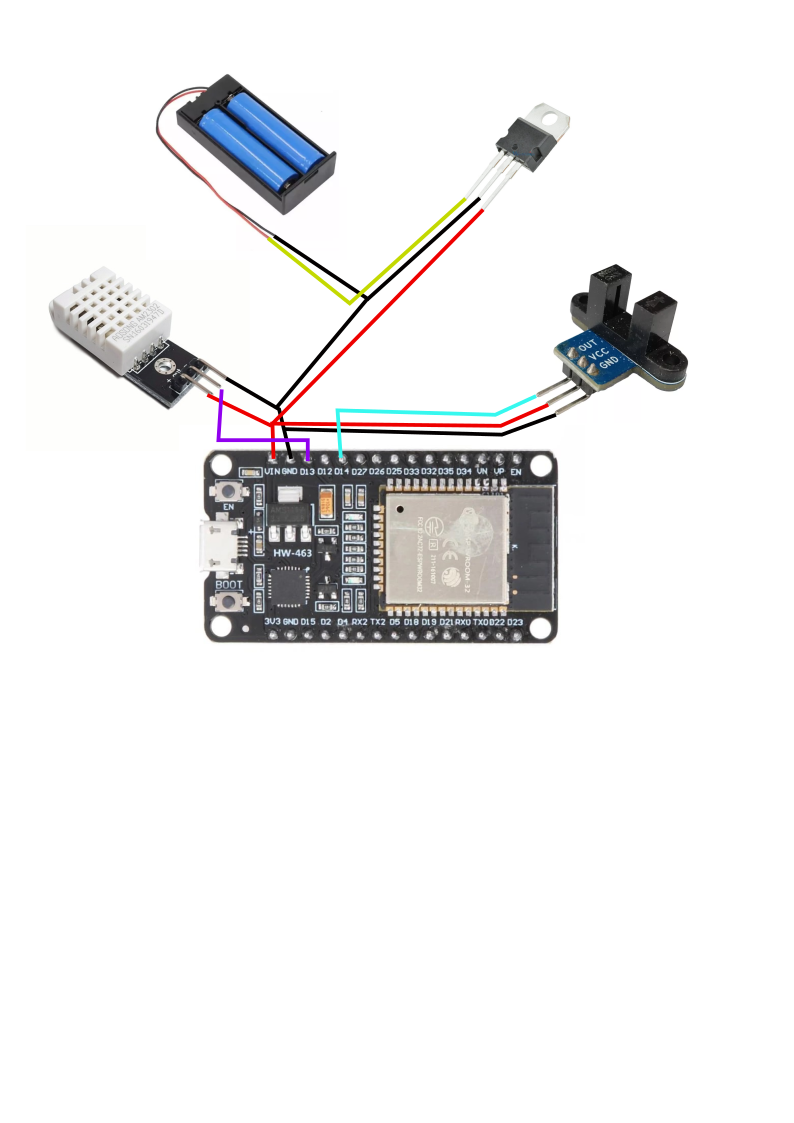
Este sensor posee tres pines GND para la conexión a tierra, VCC para la conexión a 5 voltios y OUT para la salida, esta salida estará conectada al pin 14 del ESP32 la cual al recibir un estado lógico ya sea HIGH o LOW desatara una interrupción para incrementar un contador, el tipo de estado dependerá del sensor y de la conexión que este posea ya sea una conexión PULL-DOWN o PULL-UP, al no poseer el modulo como tal, este sensor se puede conectar con una resistencia a GND o VCC dependiendo del estado que se desee obtener para realizar el conteo.

|  |  |  |  |
| --- | --- | --- | --- |
| Dispositivo | Pines Dispositivo | Pines ESP32 | Descripción |
| DHT22 | GND | GND | Conexión a tierra |
| VCC | VIN | Conexión a 5V |
| OUT | 13 | Entrada de datos |
| Sensor de barrera | GND | GND | Conexión a tierra |
| VCC | VIN | Conexión 5V |
| OUT | 14 | Entrada de datos |

***Tabla 1: Resumen de las conexiones de los dispositivos***

### Alimentación

Todo el sistema esta alimentado mediante dos baterías recargables de 3.7 voltios de litio conectadas en serie, en dicha conexión su voltaje supera el voltaje que el ESP32 y los sensores permiten por lo que además se incorpora un integrado **LM7805** el cual reduce y mantiene estable el voltaje a 5 voltios los cuales están conectados a las entradas VCC de los sensores y al pin de VIN del ESP32, este integrado posee la desventaja de ser poco eficiente puesto que para regular el voltaje la energía restante la transforma en calor por lo que es necesario incorporar un pequeño disipador al integrado, como futura mejora se podría incluir un módulo **STEP DOWN** el cual permite bajar el voltaje conmutando una entrada y obteniendo una salida ajustada, además de incorporar un modulo **BMS** el cual controle la carga y descarga de las baterías, esto para alargar su vida útil, sin embargo con el sistema implementado solo de baterías y el regulador **LM7805** el sistema puede estar en funcionamiento aproximadamente 36 horas según las pruebas realizadas.

******

***Figura 3: Diagrama de conexiones del sistema***

## CÓDIGO

Antes de comenzar el con el código es recomendable crear un nuevo proyecto en Arduino IDE y asignarle un nombre, al hacer esto se creará una carpeta con el mismo nombre del sketch.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

***Figura 4: Proyecto creado en Arduino IDE***

Una vez creado el proyecto será necesario crear un archivo con la extensión **“.h”,** llamado **“Datos”** este archivo contendrá la información sensible del proyecto, como nombre de la red WIFI, contraseña, nombre del proyecto de Blynk y otros parámetros que serán necesarios para la conexión del sistema, para crearlo se dará clic en el icono con tres puntos en la esquina superior derecha por debajo del icono del monitor serial y se seleccionará **New Tab ó nueva pestaña.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

***Figura 5: Desplegable para crear un nuevo archivo dentro del proyecto***

Al hacer clic en New Tab aparecerá un cuadro diálogo en el cual se pedirá ingresar el nombre del nuevo archivo y su extensión, este tendrá el siguiente nombre **“Datos.h”,** posteriormente se le dará clic en OK.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

***Figura 6: Cuadro de diagolo para crear el nuevo archivo***

En este nuevo archivo situaremos todos los datos sensibles del proyecto incluida la información de conexión con Blynk que se rellenara más adelante. El archivo tendrá las siguientes líneas de código:

Imagen que contiene Tabla

Descripción generada automáticamente

***Figura 7: Código del archivo Datos.h***

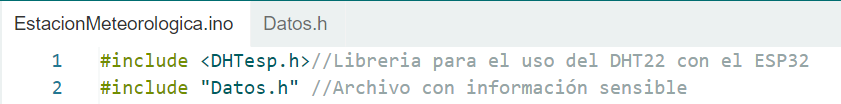
Las primeras tres líneas de código proporcionan información para conectarse al servicio de IoT en la nueve Blynk, estos datos serán proporcionados al crear un proyecto en Blynk procedimiento que se verá más adelante, también se encuentran más detallados en la guía 3 de ESP32.

En la línea 5 se incluye la librería que posee el código para interactuar con la plataforma de Blynk, esta librería se puede instalar desde la página oficial o siguiendo los pasos de la guía 3. Seguidamente en las últimas tres líneas se colocarán los datos correspondientes a la conexión a la red WIFI, estos datos serán el SSID y la contraseña del Reuter al cual se desea que el ESP32 se conecte, si es necesario agregar nuevas funcionalidades al sistema que requieran de otros datos sensibles como contraseñas para realizar actualizaciones de código por medio OTA u otra funcionalidad se recomienda incluir esta información en este archivo al compartirlo con alguna otra persona se recomiendo solo entregar la plantilla antes mostrada por motivos de seguridad.

**Nota: Se deberá sustituir los guiones por la información individual.**

En archivo **“.ino”** se encontrara toda la lógica del sistema los cuales se irán detallando paso a paso a continuación:

Primero será necesario incluir las librerías y archivos necesarios, en este caso se incluirá la librería **DHTesp.h,** la cual servirá para controlar de manera eficiente el modulo sensor de temperatura y humedad **DHT22,** esta librería se puede instalar desde el gestor de librerías, seguidamente se incluirá el archivo con los datos sensibles antes creado llamado **“Datos.h”** de la misma forma que se incluye cualquier librería.



***Figura 8: Instalado librerías***

A continuación se realizara la declaración de las variables necesarias para el funcionamiento del sistema, primero se declarara una variable que tendrá el nombre **contador** esta variable permitirá almacenar el conteo de cada vez que el sensor de barrera detecte un diente y posteriormente se utilizara para realizar cálculos, esta variable es de tipo **Volátil** ya que no deseamos almacenar siempre su valor solo mientras estemos realizando el conteo, al declararla de esta forma se almacenara en la memoria RAM del ESP y no en la memoria interna, luego se creara una variable de tipo **double** llamada **Velocidad** en la cual se almacenara el valor de la velocidad del viento que se calculó.

Una vez hecho esto se declarara una variable llamada **pi** la cual almacenara el valor de pi y será usado para realizar cálculos, además se creara otra con el nombre **dhtPin** la cual almacenara el valor del pin al cual estará conectado el módulo DHT22, este pin será el número **13 del ESP32** y por ultimo se declara una variable llamada **factor** esta variable almacenara un número el cual servirá como factor de error para corregir la velocidad del viento y así aumentar su precisión, ya que el sistema por los componentes mecánicos puede no ser tan preciso se deberá ajustar este valor dependiendo de las mediciones realizadas y comparándolas con los valores de un anemómetro real, por defecto estará en 2.2.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

***Figura 9: Declaración de variable***

Antes de comenzar con **el void setup** será necesario instanciar la clase DHTesp la cual está dentro de la librería que controla el módulo DHT22.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

***Figura 10: Instancias***

Dentro del **Void Setup** colocaremos algunos valores iniciales, primero realizaremos la conexión a la red WiFi y a los servidores de Blynk, estos dos procesos los realiza de manera automática una función que trae la librería de Blynk llamada **“begin()”,** esta recibe tres parámetros, el token para la conexión con Blynk, el SSID del Reuter al cual sea desea conectar y la contraseña, estos datos serán colocados en el archivo independiente llamado **“Datos.h”** y únicamente se nombrara su variable correspondiente.

Seguidamente se configurara la librería para controlar el módulo DHT22, esta librería posee el nombre **“setup()”** y recibe dos parámetros, el primero es el pin al cual estará conectado el modulo el cual fue declarado por medio de la variable **dhtPin**, como segundo parámetro se indicara que sensor se utilizara ya que existe un DHT11 el cual es una versión similar pero con otras características y el DHT22, en este caso se indicara el segundo colocando **“DHTesp::DHT22”.**

Además se utilizara el led incorporado del ESP para indicaciones de usuario, este led esta asociado al pin número 2 por lo que se declara este pin como una salida por medio de la función **“pinMode()”,** seguidamente se iniciara el monitor serial con una velocidad de 57600 baudios.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

***Figura 11: Inicializaciones***

También es necesario controlar el sensor de barrera, este sensor funcionara por medio de una interrupción que será detectada en el pin 14, esta interrupción será del tipo **RESSING** lo cual indica que se detectara cuando suceda la transición entre un estado alto a un estado bajo y viceversa, la interrupción estará controlada por medio de una función llamada **“interrupcion0”** la cual será declarada más adelante.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

***Figura 12: Declarando la interrupción***

Ahora es necesario comprobar la conexión con el servidor de Blynk, para saber el estado de conexión se utilizara el led que esta integrado con el ESP32 en el pin 2, se encenderá el led y se apagara hasta que se verifique que la conexión al servidor de Blynk es exitosa, para verificar esta conexión se utilizara un ciclo **While()** el cual se ejecutara mientras la condición dada no cambie, la condición que recibirá será **“!Blynk.connect()”**, esta función devuelve verdadero si la conexión con Blynk se realizó, por lo tanto el ciclo se ejecutara mientras se encuentre desconectado por medio del operador de negación **“!”**, dentro solo imprimirá en el monitor serial **“Conectando!”.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

***Figura 13: Verificando conexión con Blynk***

Dentro del **Void Loop()** lo primero será ejecutar Blynk, para poder utilizar sus funciones y que este se encargue de estar en constante comunicación del servidor, esto se hará con la función **“run()”**, seguidamente se comprobara cada ciclo del Loop si se esta conectado al servidor, en caso contrario se encenderá el led que viene incorporado con el ESP32, utilizando la función **Blynk.connected(),** esta función de forma similar a la anterior retorna verdadero o falso si ya se tiene o no una conexión con los servidores de Blynk.

Texto

Descripción generada automáticamente

***Figura 14: Iniciando ejecución y comprobación de conexión***

Si se tiene una conexión con Blynk se procederá a realizar toda la lógica del sistema.

Primero se detectara la temperatura y la humedad que esta registrando el sensor DHT22, esto se hará por medio de la función **“getTempAndHumidity()”** el cual devuelve un objeto que se guardara en una variable llamada **“datos”** del tipo **“TempAndHumidity”** es un tipo especial de variable que permite guardar objetos de la clase **TempAndHumidity,** se esperara por medio de un delay un tiempo para realizar la lectura.

Texto

Descripción generada automáticamente con confianza media

***Figura 15: Leyendo la Temperatura y la Humedad***

Luego se procederá a calcular la velocidad del viento, para hacerlo es necesario utilizar la formula del perímetro de un círculo la cual es **“2\*pi\*r”** seguidamente se le multiplicara el **“contador”** la cual es la variable que guarda el conteo de los dientes que detecta el sensor de barrera infrarrojo por medio de las interrupción del pin 14, este valor se dividirá por el total de dientes en el disco el cual corresponde a **60 dientes** y así se obtendrá el valor aproximado de la velocidad del viento, sin embargo pueden haber problemas mecánicos que generen más fricción o rozamiento por lo que puede afectar a la lectura precisa del anemómetro, por eso a este ultimo resultado se le multiplicara un **factor** de error el cual fue declarado al inicio como **factor** el cual tiene el valor por defecto de **2.2.**

Seguidamente se imprimirán en el monitor serial los datos de temperatura, humedad y velocidad del viento.

Texto

Descripción generada automáticamente

***Figura 16: Calculando velocidad del viento***

Ahora solo queda enviar los datos a los servidores de Blynk, esto se logra utilizando la función de **“virtualWrite()”,**  ya que se especificarán estos datos como pines virtuales en Blynk, **(Para más detalle leer la guía 3),** esta función recibe dos valores el primero es el nombre del pin asociado al dato que se desea enviar y el segundo el valor que se mandara, esto para los pines de Temperatura, Humedad y velocidad del viento.

Para obtener los valores de temperatura y humedad que se almaceno en la variable **“datos”** se utilizaran las variables **“temperatura” y “humidity”** respectivamente las cuales están dentro del objeto que se almaceno en **“datos”.** Luego para finalizar el código será necesario reiniciar el contador para luego generar una nueva lectura.

Texto

Descripción generada automáticamente

***Figura 17: Enviando datos a Blynk***

Por último, fuera del void loop será necesario crear la función **“interrupcion0”,** la cual es la que se encargara de incrementar el contador cada vez que se genere una interrupción por parte del sensor de barrera infrarrojo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

***Figura 18: Función de la interrupción***

## APLICACIÓN DE BLYNK

Para configurar la Estación Meteorológica con Blynk se seguirán los siguientes pasos:

1. Se creará una cuenta de Blynk y se iniciará sesión, para más detalle ver la guía 3.
2. Una vez dentro de la consola de Blynk en el navegador web, se seleccionará el apartado de **Developer Zone** que se encuentra en la barra lateral izquierda, seguidamente se seleccionará **My Templates** y se dará clic **New Template.**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***Figura 19: Creación de la plantilla***

1. Una vez se da clic en New Template se desplegará un cuadro de diálogo en el cual se pedirá ingresar el nombre, el dispositivo a utilizar en este caso el ESP32 y la conexión del dispositivo la cual será WIFI, también se puede agregar una pequeña descripción del proyecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

***Figura 20: Configuración de la plantilla***

1. Una vez creada la plantilla se muestra en un recuadro negro en la parte derecha de la pantalla información de la plantilla, en ella se encontra el **BLYNK\_TEMPLATE\_ID** esta línea se copiará y se pegara en la línea correspondiente del archivo **Datos.h** que se creó anteriormente.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***Figura 21: Información de la plantilla creada***



***Texto

Descripción generada automáticamente con confianza media***

***Figura 22: Pegando el BLYNK\_TEMPLATE\_ID***

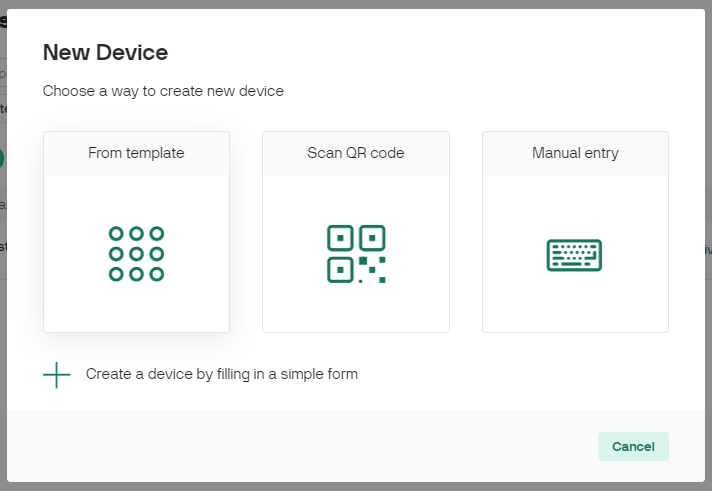
1. Luego, se creará un dispositivo para asociarla a la plantilla creada anteriormente, este hará en la opción **“Devices”** de la barra izquierda y se dará clic en el botón **“New Device”.**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***Figura 23: Creando un nuevo dispositivo.***

1. Al dar clic en New Device se desplegará un cuadro en el cual se seleccionará la opción **“From template”.**



***Figura 24: Opciones para crear dispositivo***

1. Al seleccionar esa opción se desplegará otro cuadro en cual se pedirá seleccionar la plantilla que se desea utilizar, se seleccionará en el primer campo el nombre de la plantilla antes creada, esto cambiará automáticamente el nombre del dispositivo.

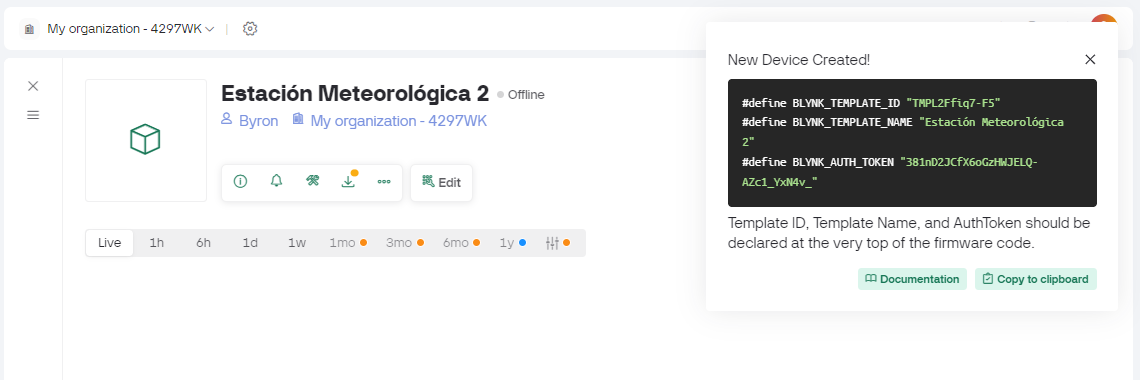
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

***Figura 25: Seleccionando la plantilla a utilizar***

1. Una vez creado el dispositivo se mostrará un recuadro negro en la esquina superior derecha con la información de acceso al dispositivo, esta información será de utilidad para conectar el ESP32 con Blynk, se copiará al porta papeles.







***Figura 26: Dispositivo ya creado***

1. Esta información copiada al portapapeles se pegará en el archivo que se creo en el proyecto de Arduino llamado **“Datos.h”** corresponde a las primeras 3 líneas del archivo.

Texto

Descripción generada automáticamente

***Figura 27: Rellenando información de Blynk***

1. Ahora es necesario crear los **Datastreams** estos son canales donde se recibirá la información de los diferentes valores que detecte el ESP32, se creará uno para la temperatura, humedad y para la velocidad del viento (Para más detalles leer la guía 3).

Para esto se seleccionará **Developer zone** y se seleccionara la plantilla que se creó al inicio, luego se dará clic en el botón **Edit.**

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente**



***Figura 28: Editando la plantilla***

1. Una vez se le de clic a **Edit,** se seleccionará el botón de **Datastream,** este esta en la barra izquierda que se encuentra en medio de la barra izquierda de la pantalla y la información de la plantilla, luego se dará clic en el botón que esta debajo del icono de la casa.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente



***Figura 29: Área de Datastreams***

1. Daremos clic en **New Datastream,** al hacer clic saldrá un menú desplegable con los diferentes tipos de pines que se pueden declarar (Para más información leer la guía 3), en este caso se usara **Virtual Pin,** se selecciona esa opción.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

***Figura 30: Creando un nuevo Virtual Pin***

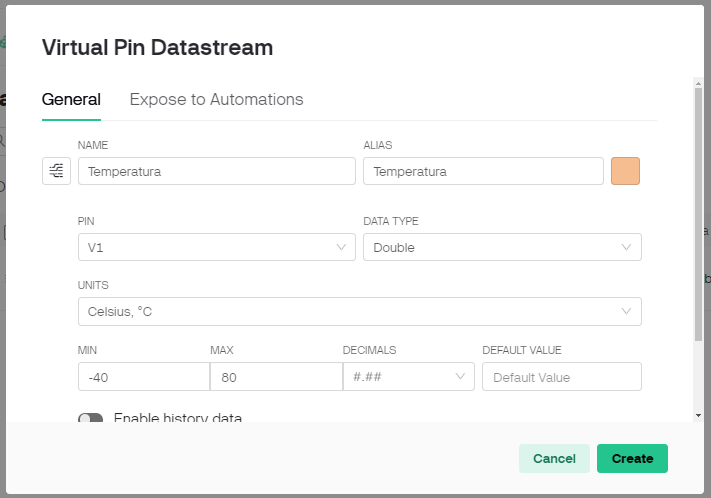
1. Al dar clic en **Virtual pin** se desplegara un nuevo cuadro de diálogo en el cual se configurara el pin, en el campo de **neme**, se colocara el nombre que tendrá el virtual pin, en este caso corresponderá a **Humedad**, el campo **Pin** se dejara en **V0**, en el campo **Data Type** se seleccionara **Double,** en **Units** se buscara **Percent** o porcentaje y en los campos **Min y Max** será colocara 0 y 100 respectivamente.

Interfaz de usuario gráfica, Tabla

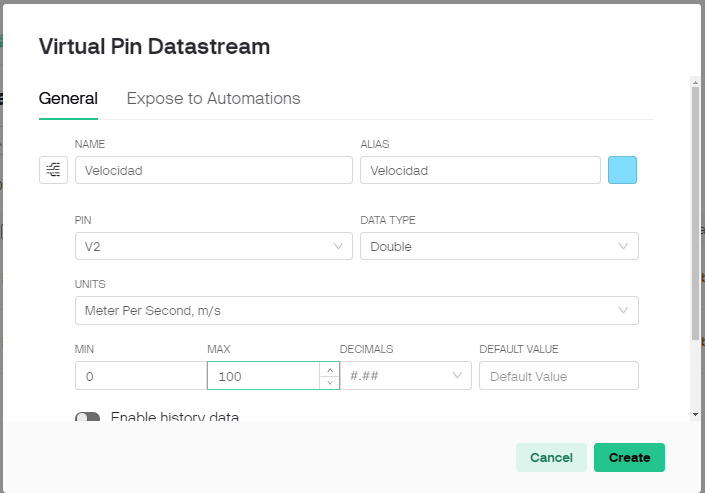
Descripción generada automáticamente con confianza media

***Figura 31: Configuración del pin para humedad***

1. Se agregarán los otros dos Virtual Pin de la misma manera, los virtual Pin para la temperatura y velocidad del viento quedarían de la siguiente forma:



***Figura 32: Configuración del Virtual pin de Temperatura***



***Figura 33: Configuración del Virtual pin de Velocidad***

1. Luego se dará clic en el botón **Save and Apply**, Seguidamente será necesario crear una interfaz en donde se puedan visualizar los datos del ESP32, esto se hace en **Web Dashboards**, para acceder a esta opción se seleccionará el icono que esta por debajo del icono de **Datastreams,** Luego se dará clic en el botón **Edit** del lado superior derecho.

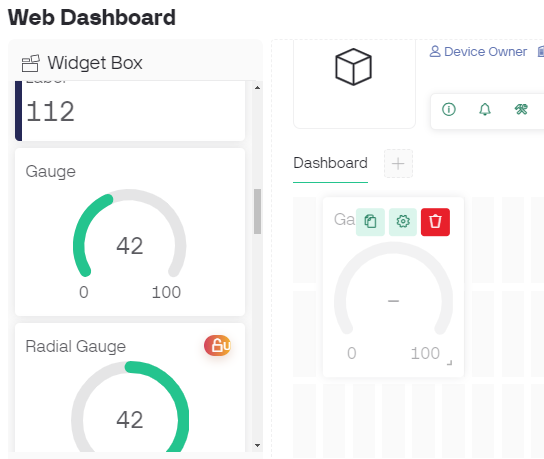
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



***Figura 34: Botón Web Dashboard***

1. Del menú de elementos con el nombre **Widget Box,** se seleccionara el elemento **Gauge** y se colocará en el área de la derecha, al hacerlo se dará clic en el icono del engranaje sobre el elemento para configurarlo.





***Figura 35: Colocando elemento***

1. Al dar clic en el icono del engranaje se desplegará un cuadro de diálogo, en el campo **Title** se colocará **Humedad** y en el campo de abajo se seleccionara el datastream asociado a la **Humedad** que se creó anteriormente, luego se dará clic en **Save.**

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

***Figura 36: Configurando Datastream***

1. Para los datos de la temperatura y la velocidad, se indicarán por medio de etiquetas, para crear una del menú de **Widget Box** se seleccionara **Label**, seguidamente se arrastrará el elemento al área y se dará clic en el icono del engranaje para configurarlo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



***Figura 37: Agregando un nuevo elemento***

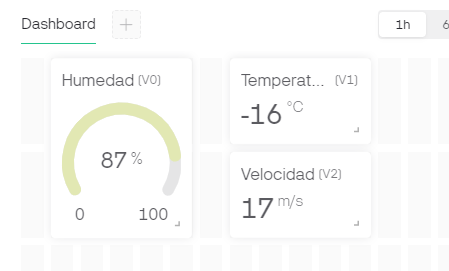
1. Similar al elemento pasado se le asignará un nombre en este caso **Temperatura** y luego se seleccionará el **Datastream** correspondiente a la temperatura, luego se dará clic en save.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

***Figura 38: Mostrando Temperatura***

1. Se realizará este mismo procedimiento para agregar la etiqueta de velocidad quedaría de la siguiente forma:



***Figura 39: Dashboard finalizado***

Por último, se dará clic en **Save And Apply,** con esto ya se estaría listo para recibir los datos.

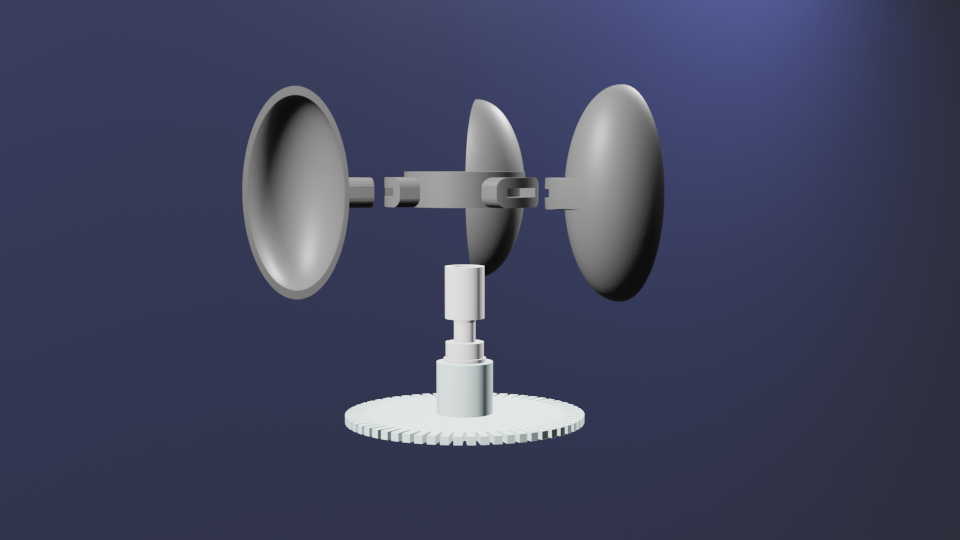
## DISEÑO 3D

Para este sistema se diseño una estructura en 3D que se compone de varios elementos, la estructura fue impresa por medio de una impresora FMD con material PLA de tres colores diferentes, Blanco, Gris y Azul traslucido, las partes se detallan a continuación:

### HELICES Y DISCO DENTADO

Para realizar el conteo por medio del sensor de barrera se diseño un disco de 60 mm de diámetro, 2 mm de espesor el cual posee 60 dientes, este a su vez posee un acople el cual a su vez esta conectado a un segundo acople, este segundo acople este sujeto a un rodamiento pequeño de 1cm, por esta razón es necesario dos acoples, el acople que está unido al rodamiento va conectado a un eje el cual puede ser un palo de pincho o un objeto similar, este eje esta unido a un soporte para hélices el cual va conectado por medio de paletas de madera a tres hélices cóncavas.

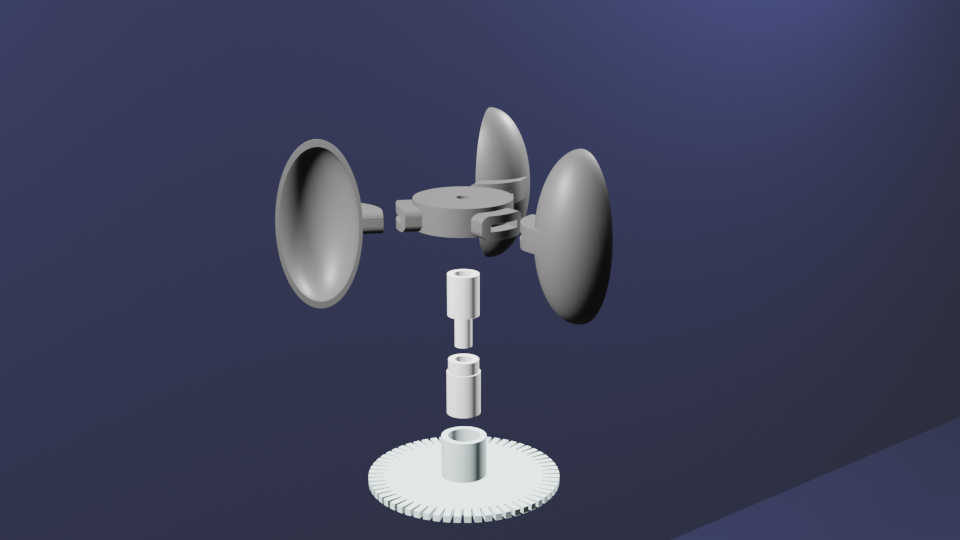
Las Hélices cóncavas permiten obtener mejor la fuerza del viento aprovechando su velocidad para impulsarse captando así más empuje.



***Figura 40: En la imagen se muestran el disco dentado, los dos acoples, la base de las hélices y las tres hélices.***

|  |  |
| --- | --- |
| **Pieza** | **Descripción** |
| 1 | Disco dentado |
| 2 | Acople de disco |
| 3 | Acople de rodamiento |
| 4 | Base de Hélices |
| 5 | Hélices |

***Tabla 2: Resumen de los elementos de Hélices***

******

**4**

**5**

**3**

**2**

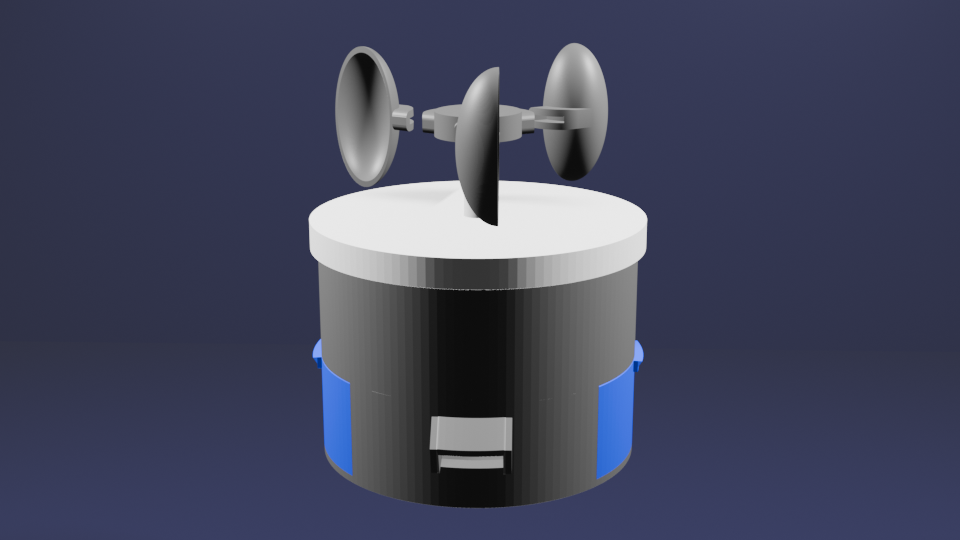
**1**

***Figura 40: Resumen de los elementos de Hélices***

### TAPADERA Y RECIPIENTE

Todo el sistema a excepción de las hélices y su base están dentro de un cilindro de 112 mm de diámetro con una altura de 90 mm aproximadamente, este cilindro posee dos partes divididas por una pared circular en el medio, en la mitad de arriba se encuentra el disco dentado, los acoples, la tapadera y el sensor de barrera, por la parte de abajo se encuentra la electrónica necesaria, el ESP32, baterías, reguladores etc. La parte inferior cuenta con dos compuertas laterales las cuales facilitan al acceso a la electrónica y las Batería.

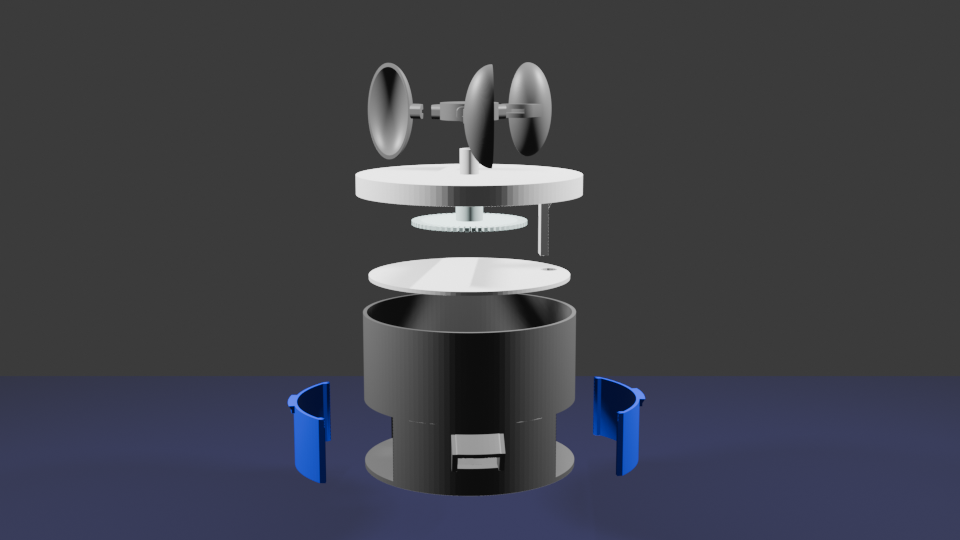
Ambas mitades están divididas por una pared circular de 2 mm, la cual puede aislarse con silicón ya que se podría usar el sistema a la intemperie, además de poseer un orificio en donde se puede colocar el módulo DHT22 para exhibirlo a la intemperie.



***Figura 41: Vista lateral del recipiente***

|  |  |
| --- | --- |
| **Pieza** | **Descripción** |
| 1 | Compuertas |
| 2 | Recipiente |
| 3 | Pared divisora |
| 4 | Tapadera |
| 5 | Sistema de Hélices |

***Tabla 3: Resumen de los elementos del Sistema***

******

5

4

3

2

1

***Figura 42: Resumen de los elementos del Sistema***

## RECURSOS