Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Querétaro

Documentación - Go Life!

Manuel Villalpando Linares

A01352033

Ian Joab Padrón Corona

A01708940

Francisco Couttolenc Ortiz

A01754959

Fecha de entrega: 29 de noviembre de 2022

Implementación de internet de las cosas (Gpo 501)

Introducción al proyecto

“No hay que ser un experto”

- Sobre Go life!

Somos una organización que se preocupa por las plantas, procurando su bienestar y se apasiona por adentrarse aún más a esas personas que muestran interés por tener plantas, es por eso que nosotros decidimos desarrollar un método de ayuda eficiente en la manera de cultivar plantas, en el cual ni siquiera se requiere conocimiento previo acerca de su humedad, temperatura o ubicación ideal.

- Misión

Buscamos apoyar principalmente la ODS 15, Vida de ecosistemas terrestres, de modo que las personas puedan comprar plantas teniendo la certeza de que habrá una aplicación web de monitoreo en vivo de sus plantas para tener noción en tiempo real de si es que les hace falta algo y poder atenderlo lo más antes posible.

- Visión

Queremos un mundo lleno de plantas, en el cual no se necesite ser un experto en botánica como dice nuestro lema, para poder comenzar a tener tu propio jardín o espacio zen dentro de tu hogar.

- Valores

Somos guiados principalmente por el amor a las plantas y tenemos noción de la importancia que tiene la ecología no solo en el medio ambiente sino la manera en la que influye dentro de nuestras vidas incluso psicológicamente, creando espacios seguros donde las personas se sientan en paz.

- Propósito

Promover que la gente tenga plantas, sin importar que no tenga la información exacta acerca de las necesidades que requiere plantar algún cultivo como chiles por ejemplo.

Creación de base de datos en MySQL

Primero que nada, para la creación de nuestra aplicación web, se debe tener definida una base de datos, para esto utilizaremos la aplicación de XAMPP, para poder hacer que nuestra computadora funcione como un servidor local al cual estaremos subiendo la información. Posteriormente iniciaremos el phpMyAdmin con el usuario y contraseñas predeterminadas de la aplicación, en este caso el usuario es “root” y la contraseña es un espacio en blanco “ “ (Si es que lo llegase a pedir).
Ya dentro del localhost, posterior al razonamiento de cómo es que irá estructurada, (¿Con qué llaves principales o foráneas cuenta? ¿Cuáles son sus tablas? ¿Cuáles son sus columnas?) pasaremos a crear la base de datos con ayuda del lenguaje SQL.

```
CREATE DATABASE maceta;

USE maceta;

CREATE TABLE planta(
	id INT(5) NOT NULL,
	date DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
	Nombre VARCHAR(15) NOT NULL
	Temperatura DECIMAL(3, 2) NOT NULL,
	Humedad DECIMAL(3, 2) NOT NULL,
	Ubicacion VARCHAR(6),

	PRIMARY KEY (id, Nombre)
);

CREATE TABLE medidas(
	Conteo INT AUTO_INCREMENT,
	Dia DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
	Hora TIME NOT NULL
	NombreP VARCHAR(15)  NULL,
	Temp_actual DECIMAL(3, 2) NOT NULL,
	Humed_actual DECIMAL(3, 2) NOT NULL,

	PRIMARY KEY (Conteo),
	CONSTRAINT nombre FOREIGN KEY (NombreP) REFERENCES planta(Nombre)
);
```

Al principio únicamente le indicamos al mySQL que queremos crear una nueva base de datos, en este caso con el nombre de maceta, por lo cual, seguido de eso se tiene que indicar que estaremos trabajando con esa base de datos

Configuración del NodeMCU

Para la realización de este proyecto, ocupamos el CHIP-INTEGRADO: ESP8266. Con este chip, podremos recopilar datos de temperatura y humedad (requeridos para el proyecto) y enviarlos a una base de datos.

Característica	ESP8266
Procesador	Tensilica LX106 32 bit a 80 MHz (hasta 160 MHz)
Memoria RAM	80 kB (40 kB disponibles)
Memoria Flash	Hasta 4 MB
ROM	No
Alimentación	3.0 a 3.6 V
Rango de temperaturas	-40°C a 125°C
Consumo de corriente	80 mA (promedio). 225 mA máximo
Consumo en modo sueño profundo	20 uA (RTC + memoria RTC)
Coprocesador de bajo consumo	No
WiFi	802.11 b/g/n (hasta +20 dBm) WEP, WPA
Soft-AP	Sí

Además, utilizamos un sensor de humedad DHT-11 de tres terminales, con las siguientes características:

MODELO	**DHT11**
Alimentación	de 3,5 V a 5 V
Consumo	2,5 mA
Señal de salida	Digital
Temperatura	
Rango	de 0ºC a 50ºC
Precisión	a 25ºC ± 2ºC
Resolución	0.1ºC
Humedad	
Rango	de 20% RH a 90% RH
Precisión	entre 0ºC y 50ºC ± 5% RH
Resolución	1% RH

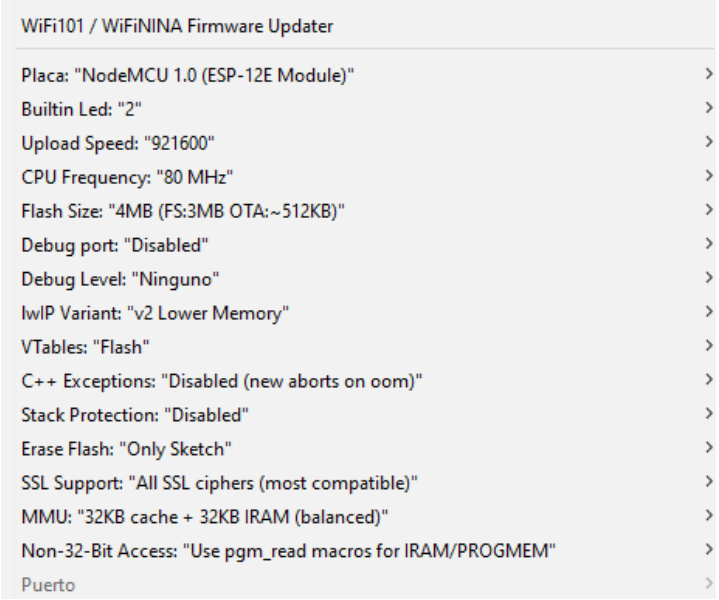
Consideraciones

Los pasos que se mencionan a continuación servirán únicamente para el Sistema Operativo Windows, por conveniencia del equipo para la realización de este proyecto.

Para poder extender el proyecto desde una red LAN, hasta un servidor Público, hicimos uso de un servicio ‘playit.gg’ que nos permite abrir los puertos de la computadora en LAN, para que puedan ser accedidos a través de internet, como si se tratara de un servidor funcional.

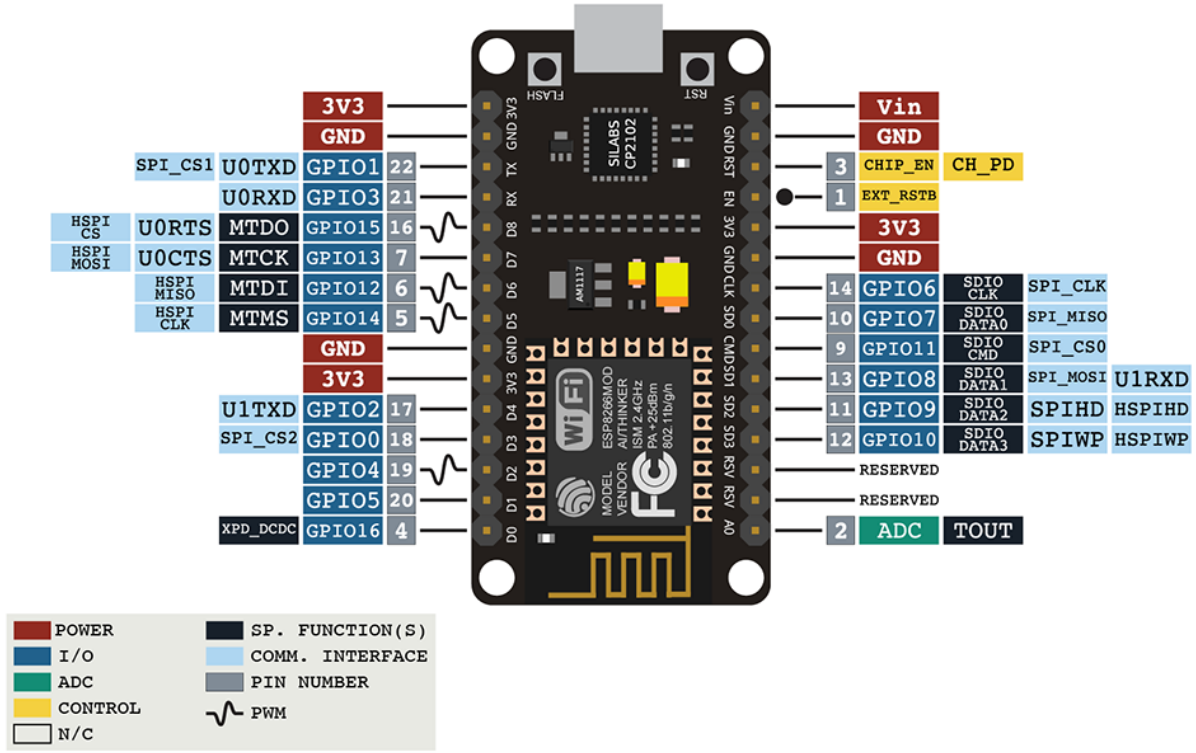
Pasos

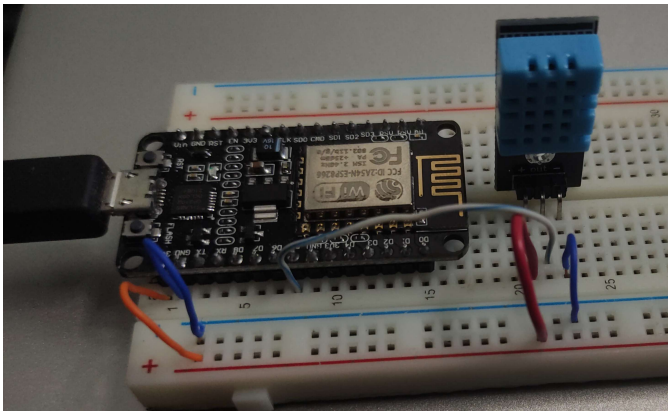
1. Descargar el Software de ArduinoIDE.
    a. Para descargar en Windows para versiones anteriores a W10: https://www.arduino.cc/en/software
    b. Para W10 y posteriores, puede ser desde la tienda de Windows: https://apps.microsoft.com/store/detail/arduino-ide/9NBLGGH4RSD8?hl=es-mx&gl=mx
2. Instalar los Drivers del NodeMCU. Si bien, en W10, al conectar el Chip a la computadora, se instala automáticamente, es mejor utilizar el archivo instalador de drivers ofrecido por: 'Silicon Labs' los creadores del micro controlador integrado al ESP8266: https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers
3. Agregar la tarjeta 'NodeMCU' al IDE de Arduino
    a. Para ello, podemos encontrar el apartado 'Aditional Boards Manager' en File>>Preferences>>Settings
    b. Utilizamos la siguiente URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json
    c. Ahora que el IDE ya cargó las librerías requeridas, vamos a instalarlas en Tools>>Board>>BoardManager y buscamos 'ESP8266'. Instalamos la última versión (Actualmente la 3.0.2)
4. Agregar la librería 'DHT sensor library for ESPx'
    a. Abrimos el gestor de Librerías con Ctrl+Shift+I
    b. Buscar la librería e instalar la última versión (autor beegee_tokyo)
5. Configurar las características específicas del NodeMCU en Arduino. Al menos para la versión que utilizamos (V2) se utilizan las siguientes características:



Alambrado del NodeMCU

No se entrará en profundidad al respecto. Sin embargo, es necesario verificar que el PIN de lectura de datos sea el correspondiente al establecido en el Código de Arduino, de acuerdo al diagrama específico del NodeMCU. Para el caso del utilizado por nuestro equipo es:





Código del Proyecto

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include "DHTesp.h"
```

```
#define HOST "iot.niubycraft2.playit.gg:6286"          //HOST URL

#define WIFI_SSID ""                                   // WIFI SSID
#define WIFI_PASSWORD ""                               // WIFI password here

#define DHTpin 14

// Variables which will be uploaded to server

float val1 = 0;
float val2 = 0;
String sendval, sendval2, postData;

DHTesp dht;                                            // Initialize DHT Sensor

void setup(){
Serial.begin(115200);              // Information transfer rate from Arduino Code to NodeMCU
Serial.println("Communication Started \n\n");
delay(1000);

dht.setup(DHTpin, DHTesp::DHT11);                      // GPIO14

pinMode(LED_BUILTIN, OUTPUT);                          // Initialize NodeMCU LED

WiFi.mode(WIFI_STA);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

Serial.print("Connecting to: ");
Serial.print(WIFI_SSID);

// Try to connect with WiFi
while (WiFi.status() != WL_CONNECTED){
   Serial.print(".");
   delay(500);
 }

Serial.println();
Serial.print("\nSuccess!");
Serial.print("\nIP Address is: ");
Serial.println(WiFi.localIP());                        // Print local IP address

delay(30);
}

void loop() {
```

```
HTTPClient http;                                    // Http object of class HTTPClient
WiFiClient wclient;                                 // Wifi-Client object of class HTTPClient

val1 = dht.getTemperature();                        // Gets the values of the temperature
val2 = dht.getHumidity();                           // Gets the values of the humidity

//Check if values are NaN or invalid
if(isnan(val1) || isnan(val2)){
 Serial.println("Error reading values from DHT Sensor");
 Serial.println("Sending 1's to Database");
 sendval = String(1);
 sendval2 = String(1);
}
else{
 // Convert float variables to string
 sendval = String(val1);
 sendval2 = String(val2);
}

postData = "sendval=" + sendval + "&sendval2=" + sendval2;

// Connect to host where dbwrite is located (PHP File to Write into SQL Database)
http.begin(wclient, "http://iot.niubycraft2.playit.gg:6286/dbwrite.php");

// Specify content-type header
http.addHeader("Content-Type", "application/x-www-form-urlencoded");

// Send POST request to php file and store server response code in variable named
httpCode

int httpCode = http.POST(postData);
Serial.println("Values to send are: Temperature = " + sendval + " && Humidity = "+sendval2 );

// Connection established with SQL Database
if (httpCode == 200){
 String webpage = http.getString();
 Serial.println(webpage + "\n");
 Serial.println("Values uploaded successfully.");
}

// Connection failed with SQL Database
else {
  Serial.println("Failed to upload values.Error code: ");
  Serial.println(httpCode);
  http.end();
  return;
}
```

```
delay(3000);
digitalWrite(LED_BUILTIN, LOW);
delay(3000);
digitalWrite(LED_BUILTIN, HIGH);
}
```

//
En este código básicamente se inicializan todas las variables necesarias para poder hacer distintas cosas:

- Leer los datos del sensor
- Crear un Objeto de tipo WiFiClient, con el cuál el NodeMCU podrá comunicarse a internet y enviar datos
- Crear un Objeto de tipo HTTPClient, para poder enviar datos via php a la base de datos
- Crear un Query (postData) con la información de temperatura y humedad que se enviarán a un php dentro del servidor, para que cargue los datos a la base de datos

Código de enlace PostData -> MySQL

//

```
<?php

// host = localhost because database hosted on the same server where PHP files are hosted
$host = "localhost";
$dbname = " ";                                  // Database name
$username = " ";                                // Database username
$password = " ";                                // Database password

// Establish connection to MySQL database
$conn = new mysqli($host, $username, $password, $dbname);

// Check if connection established successfully
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

else { echo "Connected to mysql database. "; }

// Get date and time variables
   date_default_timezone_set('America/Mexico_City');  // For other timezones, refer:-
https://www.php.net/manual/en/timezones.asia.php
   $d = date("Y-m-d");
   $t = date("H:i:s");
```

```
// If values send by NodeMCU are not empty then insert into MySQL database table

  if(!empty($_POST['sendval']) && !empty($_POST['sendval2']) ){
       $val = $_POST['sendval'];
       $val2 = $_POST['sendval2'];
       $sql = "INSERT INTO nodemcu_table(val, val2, Date, Time) VALUES
('".$val."','".$val2."', '".$d."', '".$t."')";

                    if ($conn->query($sql) === TRUE) {
                        echo "Values inserted in MySQL database table.";
                    }
                    else {
                        echo "Error: " . $sql . "<br>" . $conn->error;
                    }
}

// Close MySQL connection
$conn->close();

?>
```

//

No somos expertos en php, pero logramos entender que con este código es posible crear un Query que lea el postData que se carga desde el Código de Arduino, y este se manda como una instrucción de tipo sql a la base de datos que se encuentra en el mismo lugar que este archivo dbwrite.php (dentro del HTDocs del servidor)

Conexión y Arranque del Servidor/Código

1. Por motivos de bloqueo de puertos de nuestra Institución, al menos para la realización de este proyecto dentro de sus instalaciones, requerimos inicializar una red Hotspot para la comunicación NodeMCU -> Internet
2. Encendido de XAMPP>>Apache/MySQL
3. Encendido del Cliente de playit.gg
4. Compilar el Código en Arduino

Si todo sale bien, el servidor ya debería estar recibiendo datos desde el NodeMCU, independientemente de que el NodeMCU esté conectado a la misma red o el Código de Arduino se ejecute en el mismo Ordenador que el Servidor.

Página Web

1- Para poder realizar la página web primero necesitamos lo que va a ser la parte principal, por lo que creamos un archivo index.html en donde sera nuestra página inicial, empezamos con la clase = Hero dentro del Header que está dentro de la sección Body, dentro de la clase = Hero ingresamos una “clase = nav container” que va a ser la el contenido de la parte superior de la página en donde encontraremos el logo.

Después creamos la sección el cual es un elemento que representa una lista de items en el cual se encuentra, inicio, acerca de y comparador. Dentro de la sección para poder representar los ítems los colocamos entre la sección el cual representa un elemento de lista.

Realizamos un sección <section class= “hero_container container> en el cual implementamos un título con <h1 class= “hero_title> Aprende a cuidar las plantas del hogar </h1> , al igual que implementamos un párrafo de la manera siguiente e implementamos una clase dentro de ella para que a la hora de realizar el código en css sea más apropiado y más fácil de implementar Elige de una vez ser un experto en el cuidado de las plantas del hogar . Después creamos un hipervínculo con la ayuda de <a> en el cual implementamos un href=, ya que funciona de la siguiente manera, si el elemento a tiene un atributo href este representará un hipervínculo, realizamos una clase dentro del elemento a class=”cta” el cual va a funcionar como un botón, por lo que quedaría de la siguiente manera: Aprende ahora, colocamos el signo “#” para que no dirija a una liga, sino que esté ligada al botón que colocamos.

//

index.html > html > body > header.hero > nav.nav.container > div.nav_logo

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Plantas</title>
    <link rel="shortcut icon" href="./images/favicon.png.png" type="image/x-icon">
    <link rel="stylesheet" href="/css/estilos.css">
</head>
<body>

    <header class="hero">
        <nav class="nav container">

            <div class="nav_logo">
                <h2 class="nav_title">Plantas Hebáceas</h2>

            </div>

            <ul class="nav_link nav_link--menu">

                <li class="nav_items">
                    <a href="#" class="nav_links">Inicio </a>

                </li>
                <li class="nav_items">
                    <a href="#" class="nav_links">Acerca de  </a>

                </li>
                <li class="nav_items">
                    <a  class="Comparador"href="comparador.html" class="nav_links">Comparador </a>

                </li>

                <img src="./images/close.svg" class="nav_close">

```

```
            </div>
        </nav>

        <section class="hero_container container">
            <h1 class="hero_title"> Aprende a cuidar las plantas del hogar.</h1>
            <p class="hero_paragraph"> Elige de una vez ser un experto en el cuidado de las plantas del hogar.

            </p>
            <a href="#" class="cta">Aprende ahora</a>

        </section>
```

//

2- Creamos un elemento <main></main> dentro del elemento body, el elemento main va a ser el encargado de representar el contenido main de la página, dentro del elemento main vamos a crear una sección el cual va a tener una clase = “container about” el cual implementamos el elemento <a> con el elemento name y una frase el cual es utilizado para direccionar a esa sección de la página, después implementamos un elemento <h2></h2> con la clase “subtitle” el cual es para representar una sección de Heading que está conectado con el name del elemento <a>. Introducimos un párrafo en el cual implementamos una clase llamada “about_paragraph” para poder eventualmente modificarlo en css.
Creamos un el cual es un elemento que no tiene un significado especial pero representa a los hijos el cual tendrá una clase= “about_main” en el cual estará dentro el elemento article representa una composición completa o independiente en un documento, una página o una aplicación el cual tendrá una clase “about_icons” el cual dentro de esta tendrá el elemento “img” que representa una imagen y se le atribuye a la clase= “about_icon”, después agregamos un elemento <h3></h3> el cual representa una sección del heading en donde irá el nombre de la clasificación del tipo de planta.

Agregamos un botón que direccione al tipo de planta, ejemplo: “botón”= Tipo de planta anual —> y vaya hacia la sección Tipo de planta anual de la página y posteriormente agregamos un párrafo en donde pondremos la descripción de la clasificación del tipo de planta, en este caso para los tipos de plantas anuales, y repetimos el mismo proceso 2 veces más para que en total sean 3 de este tipo y así tendremos la imagen de la planta en forma de icono, el tipo de planta del que se quiere tratar y una pequeña información acerca de esta clasificación. Todo esto dentro del y < su vez dentro de <section></section>.

//

index.html > html > body > main > section.container.about > div.about_main > article.about_icons

```
62      <main>
63          <section class="container about">
64              <a name="¿Que aprenderas en esta pagina web?"> </a>
65              <h2 class="subtitle">¿Que aprenderas en esta pagina web? </h2>
66              <p class="about_paragraph">Lo necesario para ser un experto en cuidado de plantas en el hogar, en el cual nos enfocaremos <br> en la cla
67                  herbáceas el cual se divide en anuales, bianuales y perennes.<br>
68                  cada una de estas clasificaciones tienen sus propios tipos de plantas.
69
70
71              </p>
72
73              <div class="about_main">
74                  <article class="about_icons">
75                      <img src="./images/leaf.svg" class="about_icon">
76                      <h3 class="about_title">Anuales </h3>
77                      <a href="#Tipos de plantas anuales" class="cta"> Tipos de plantas anuales</a>
78                      <p style="text-align: center"class="about_paragraph">completan su ciclo en 1 año; crecen a partir de una semilla, florecen, y mu
79
80
81
82
83                  </article>
84
85                  <article class="about_icons">
86                      <img src="./images/leaf.svg" class="about_icon">
87                      <h3 class="about_title">Bianuales </h3>
88                      <a href="#Tipos de plantas bianuales" class="cta" > Tipos de plantas bianuales </a>
89                      <p style="text-align: center" class="about_paragraph">Se agrupan todas aquellas que denominamos de "temporada de otoño", que req
90
91
92
93
94                  </article>
95
```

index.html > html > body > main > section.container.about > div.about_main > article.about_icons

```
95
96                  <article class="about_icons">
97                      <img src="./images/leaf.svg" class="about_icon">
98                      <h3 class="about_title">Perennes </h3>
99                      <a href="#Tipos de plantas perennes" class="cta" > Tipos de plantas perennes </a>
100                     <p  style="text-align: center" class="about_paragraph">Aquella que vive durante más de dos años o, en general, florece y produce
101
102
103
104
105                 </article>
106
107
108
109             </div>
110
111
112
113
114
115
116         </section>
117
```

//

3- Después creamos dentro del <main></main> otra <section></section> el cual contendrá la clase “Testimony” que nos va a servir para poder crear el slash de la presentación de los integrantes del equipo dentro de la página web. Dentro del elemento que creamos <section></section> colocamos el elemento el cual tendrá la clase= “testimony_container container” que significa que todo lo que realicemos dentro de este <section></section> será el contenido.

Dentro del ingresamos una imagen por medio del elemento <img></img> el cual será el diferenciador de las flechas que se implementan, denominaremos la clase como “testimony_arrow” (que indica la imagen de la flecha) y ingresamos un “id” el cual es definido como único en todo el documento ya que su propósito es identificar el elemento a la

hora de linkear y el id lo determinamos como “ before”, por lo que nos queda <img src=”./images/leftarrow.svg” class=”testimony_arrow” id=”before”>.
Una vez teniendo esto en cuenta creamos una <section></section> dentro de la section principal el cual le agregamos una “class” y la denominamos “testimony_body testimony_body–show” y le agregamos data-id=”1”, este para diferenciar el orden de cómo van a ir pasando las imágenes y la información que se va a tener en testimony. Dentro del segundo <section></section> ingresamos un el cual contendrá la clase = “testimony_texts” y dentro del div contendrá el elemento <h2></h2> que llevara una clase denominada “subtitle” en el cual vendrá el nombre del estudiante, posteriormente en la misma línea de código abriremos un elemento con la clase = “testimony_course” el cual contendrá la leyenda que se le quiera ingresar.
Ingresamos un párrafo con el elemento el cual contendrá la clase = “testimony texts” esto con la finalidad de que a la hora de pasarlo a editarlo a css sea más práctico y fácil de editar debido a que se encuentra todos los párrafos de la sección slide dentro de una misma clase, por lo que a la hora de editarlo en css se editaran todos los párrafos que están dentro de esa sección slide, lo mismo pasa para cada una de las clases que están determinadas en cierta parte del código, para que sea más rápido y sencillo de editar cualquier aspecto de la página mediante el uso de css.

Implementamos el elemento <figure></figure> el cual representa algún contenido de flujo, opcionalmente con un titulo, que es independiente en el cual colocaremos la clase= “testimony_picture” en el cual dentro de <figure></figure> colocaremos un elemento de img el cual representa una imagen el cual ingresamos un url (src) que va a estar direccionado a la carpeta images que creamos en un inicio y se direcciona una vez estando dentro de la carpeta a la imagen que queremos seleccionar. Una vez teniendo esto en cuenta lo clasificamos con la clase = “testimony_img”.

Repetimos el mismo procedimiento 2 veces más para que sea en total 3 de estas secciones, lo único que modificaremos será el “id” cambiandolo por 2 o por 3 para diferenciar la posición en la que estarán los slides y cambiamos la leyenda de la clase = “subtitle” junto con la leyenda del elemento span, ambos se cambian, por lo que nos quedaría un slide capaz de poder moverse de izquierda a derecha y en orden para cerrar con la otra imagen de la flecha pero del lado contrario.

Conexión MySQL -> Página web mediante php

1. Para poder enlazar la base de datos a nuestra página web se necesita un método, en este caso utilizaremos php, ya que nos permite utilizar las variables como cualquier otra variable de programación dentro del código, dándonos una amplia posibilidad de acciones con la misma, desde comparar hasta utilizarla para suma o restar algo.

2. Para poder enlazar ambas cosas con php primero que nada se necesita iniciar el php dentro de la página que queremos que enlace de nuestro html, por lo que tendremos que agregar el siguiente código en la parte superior de la página en la que estaremos trabajando con nuestra base de datos:

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
<?php

  $host = 'iot.niubycraft2.playit.gg';
  $db = 'microsoft2';
  $puerto = '6286';
  $usuario = 'root';
  $password = '';

  $link = mysqli_connect($host,$usuario,$password,$db);

?>
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

   En el código se tiene hasta afuera del php, “<?php” “?>” con el código php enmedio de estos para abrir y cerrar indicando desde donde comienza hasta donde acaba.
   Después inicializamos variables de host, db, puerto, etcétera, esto para que el php pueda leerlos e inicialice en la ruta del servidor correspondiente donde estaremos cargando tanto la base de datos como la página web.
   En la última línea antes de cerrar el código se define la variable de link, uniendo las demás variables dentro de la función mysqli_connect(), la cual es una función predefinida que le permite a php inicializar con todos esos valores que ingresamos anteriormente.
3. Después de escribir el php para inicializar la página en el servidor con el puerto y las indicaciones especificadas en las variables, cerramos el php y ahora sí podemos comenzar a escribir el código de la página en la cual estaremos utilizando la base de datos de MySQL (En caso de no contar previamente con uno).
4. Dentro del html, en el body, tenemos que inicializar un nuevo texto php, esto para ahora sí definir las variables de los datos que estaremos utilizando.
5. Se ingresa la búsqueda del SQL en alguna variable del php, para poder inicializarla más tarde y se junta con el link de la base de datos con una de las funciones preestablecidas para correr el query (mysqli_query(link,query)).
6. Se le asignan variables a las columnas que estaremos utilizando.

7. Se utiliza un ciclo while con un echo para imprimir varias veces las variables en las que tengamos asignadas nuestras columnas.
8. Además de eso, en nuestro caso hicimos una prueba de una condición, para poder asignar un css posteriormente que nos permitiera cambiar de estilo si es que los valores leídos de las variables del sql son diferentes a un rango establecido por nosotros.
9. Por último se cierra el MySQL utilizando la función mysqli_close($link) cuando ya terminamos de definir los querys que estaremos utilizando a lo largo de nuestra página.
    a. Nota: Se debe recordar en todo momento que el php donde se estará utilizando y buscando los querys se inicializa dentro del html.

Ejemplo del texto anterior:

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
 <?php

        $query = "SELECT * FROM planta";
        $querytempreal = "SELECT *
        FROM medidas
        ORDER by Conteo DESC
        LIMIT 1";

        $resultado = mysqli_query($link,$query);
        $tempreal = mysqli_query($link,$querytempreal);


        while($fila = mysqli_fetch_array($tempreal)) {

            $dia = $fila['dia'];
            $nombre = $fila['NombreP'];
            $temperatura = $fila['Temp_actual'];
            $humedad = $fila['Humed_actual'];

            if (abs($tempreal-$resultado) > 5) {
                $tipo = 'fuera';
                echo 'Fuera';
            }
            echo
'<tr><td>'.$dia.'</td><td>'.$nombre.'</td><td>'.$temperatura.'</td><td>'.$humedad.'<
/td></tr>';
        }
        mysqli_close($link);
    ?>
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```