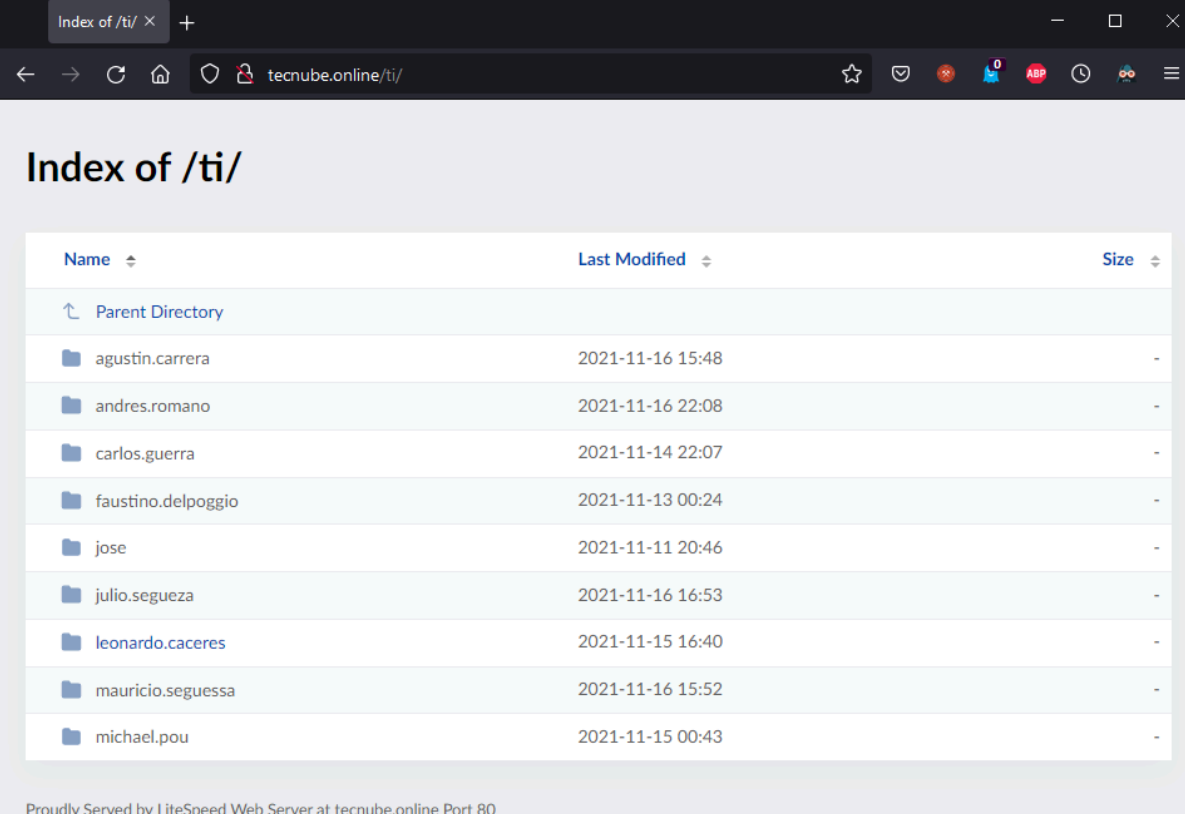


**INTRODUCCIÓN A LOS SISTEMAS DE CONTROL**  
**Integración de tecnologías para el desarrollo del**  
**módulo de control de un brazo robot mediante un**  
**servidor web en la nube.**

**16/11/2021**

Para esta actividad era necesario manejar la información de los servomotores desde un servidor web, para ello se utilizó el dominio <http://tecnube.online/ti/> provisto por el docente.

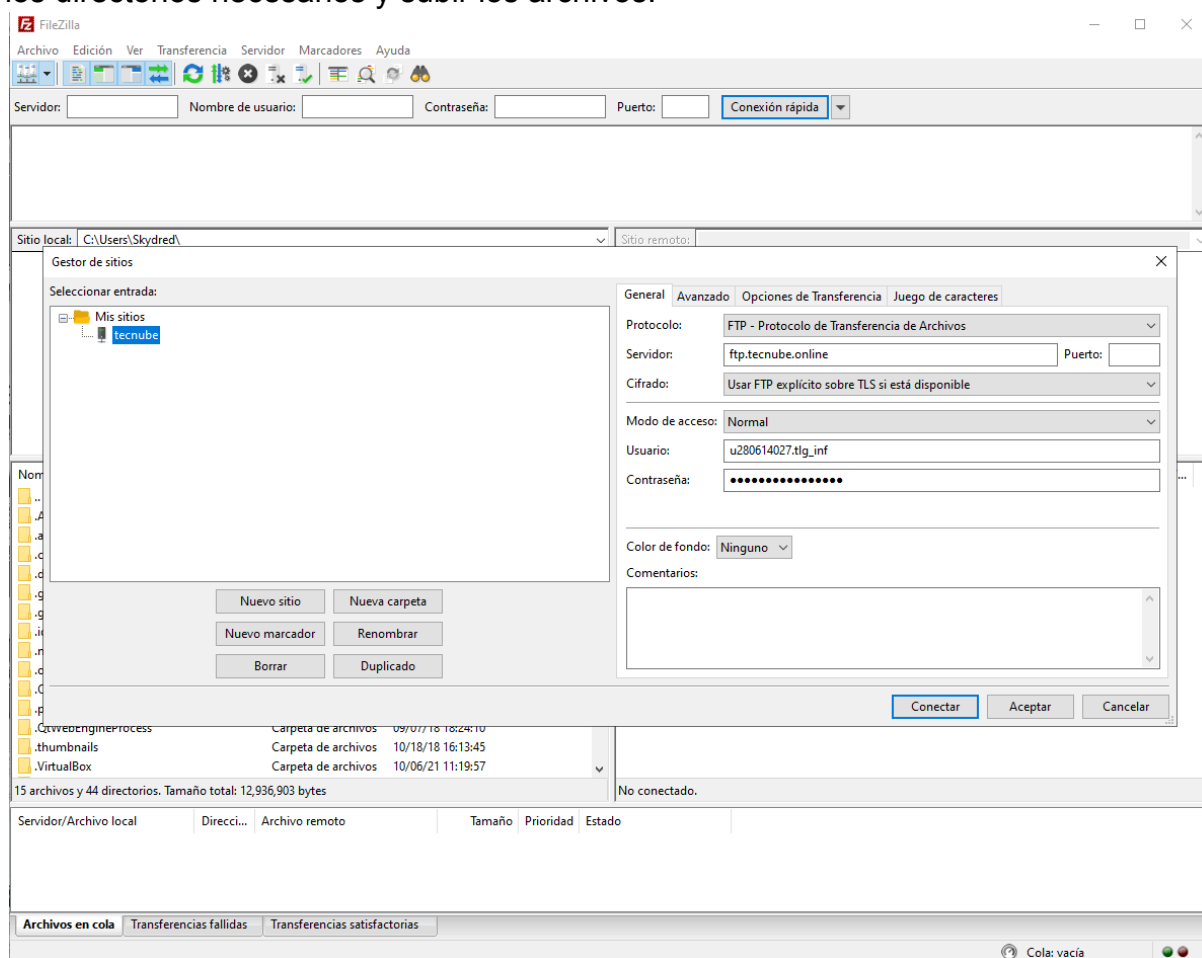


Index of /ti/

Name	Last Modified	Size
<a href="#">Parent Directory</a>		
<a href="#">agustin.carrera</a>	2021-11-16 15:48	-
<a href="#">andres.romano</a>	2021-11-16 22:08	-
<a href="#">carlos.guerra</a>	2021-11-14 22:07	-
<a href="#">faustino.delpoggio</a>	2021-11-13 00:24	-
<a href="#">jose</a>	2021-11-11 20:46	-
<a href="#">julio.seguez</a>	2021-11-16 16:53	-
<a href="#">leonardo.caceres</a>	2021-11-15 16:40	-
<a href="#">mauricio.segussa</a>	2021-11-16 15:52	-
<a href="#">michael.pou</a>	2021-11-15 00:43	-

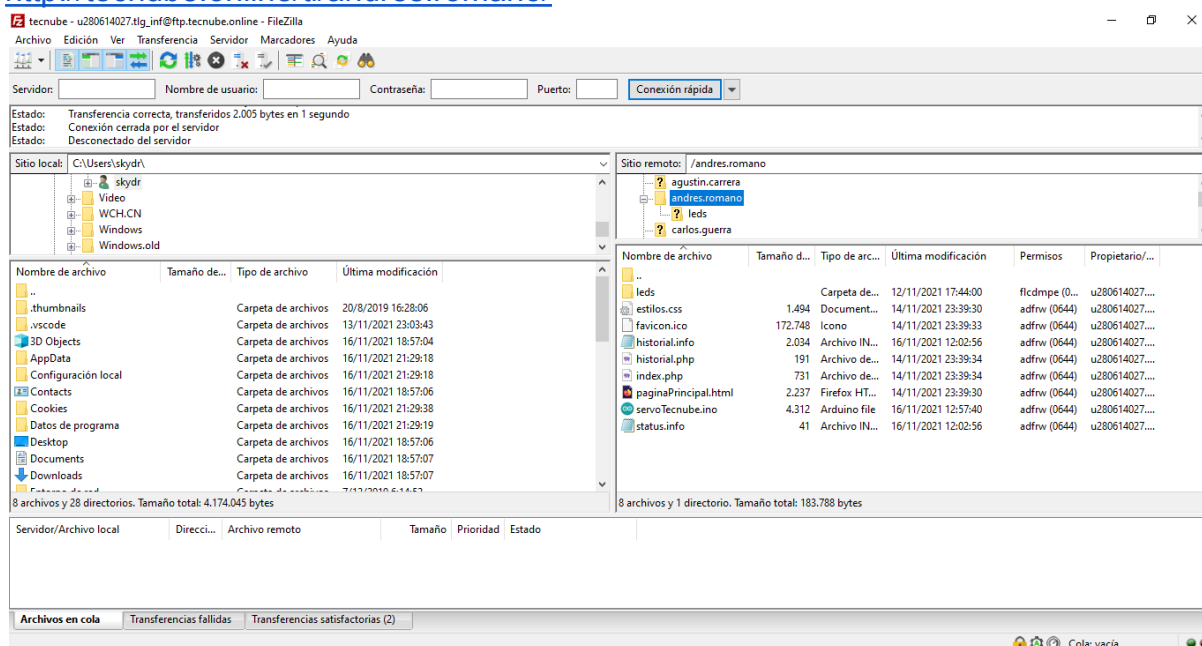
Proudly Served by LiteSpeed Web Server at tecnube.online Port 80

Desde un gestor FTP (en este caso Filezilla) se logra acceder al servidor para crear los directorios necesarios y subir los archivos.



La carpeta correspondiente a este documento es

<http://tecnube.online/ti/andres.romano/>



Para gestionar los ángulos se utilizaron los mismos archivos que para el laboratorio anterior, pero con mínimos cambios; En rojo se resaltan los códigos de la página y el azul los archivos que usaremos para guardar información (más el ícono de la página).

Sitio remoto: /andres.romano

- agustin.carrera
- andres.romano**
- leds
- carlos.guerra

Nombre de archivo	Tamaño d...	Tipo de arc...	Última modificación	Permisos	Propietario/...
..					
leds		Carpeta de...	12/11/2021 17:44:00	flcdmpe (0...	u280614027....
<b>estilos.css</b>	1.494	Document...	14/11/2021 23:39:30	adfrw (0644)	u280614027....
<b>favicon.ico</b>	172.748	Icono	14/11/2021 23:39:33	adfrw (0644)	u280614027....
<b>historial.info</b>	2.034	Archivo IN...	16/11/2021 12:02:56	adfrw (0644)	u280614027....
<b>historial.php</b>	191	Archivo de...	14/11/2021 23:39:34	adfrw (0644)	u280614027....
<b>index.php</b>	731	Archivo de...	14/11/2021 23:39:34	adfrw (0644)	u280614027....
<b>paginaPrincipal.html</b>	2.237	Firefox HT...	14/11/2021 23:39:30	adfrw (0644)	u280614027....
<b>servoTecnu.be.ino</b>	4.312	Arduino file	16/11/2021 12:57:40	adfrw (0644)	u280614027....
<b>status.info</b>	41	Archivo IN...	16/11/2021 12:02:56	adfrw (0644)	u280614027....

8 archivos y 1 directorio. Tamaño total: 183.788 bytes

Para que la página web funcione perfectamente solamente fue necesario cambiar la url del header en **index.php** por su equivalente dentro de la nube.

```

<?php
date_default_timezone_set('America/Argentina/Salta');

$ang1 = $_POST['ang1'];
$ang2 = $_POST['ang2'];
$ang3 = $_POST['ang3'];
$ang4 = $_POST['ang4'];
$ang5 = $_POST['ang5'];
$ang6 = $_POST['ang6'];

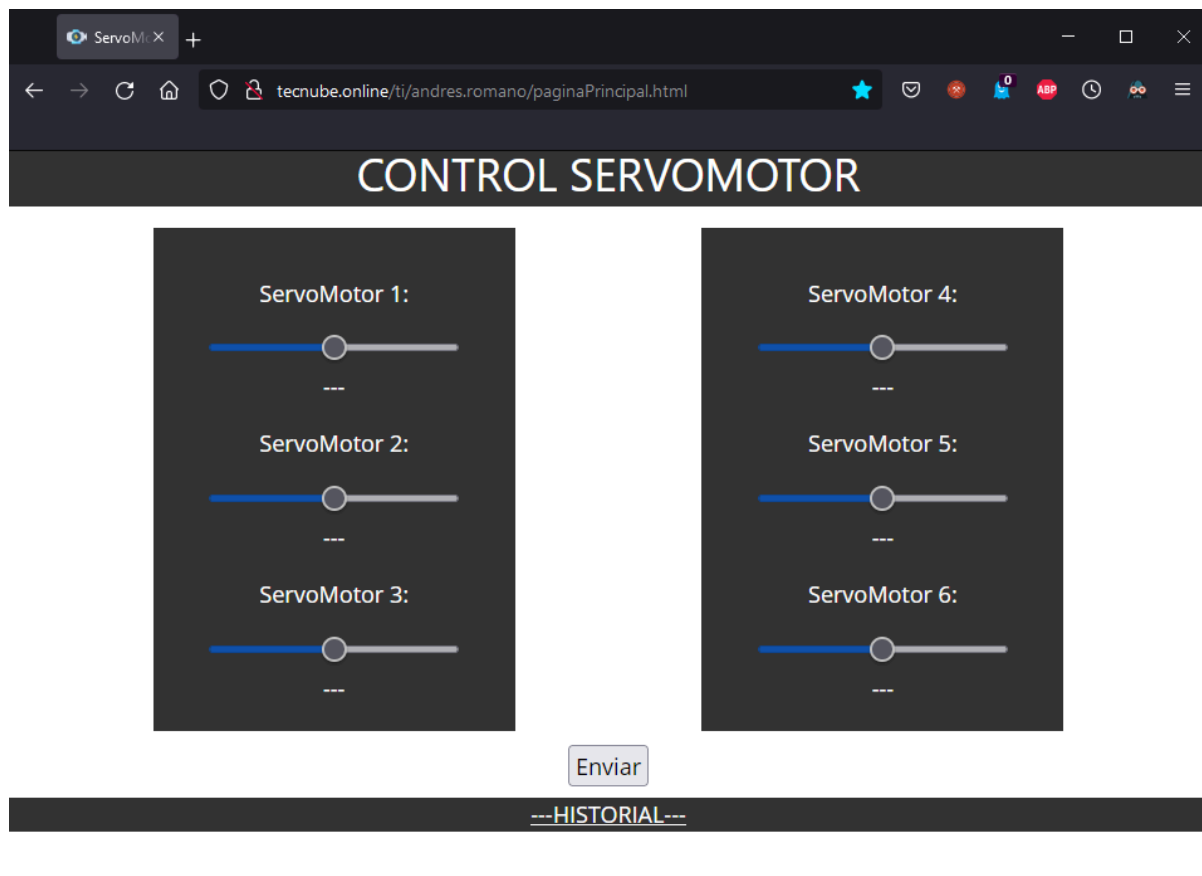
$info = "[" . date("Y-m-d H:i:s") . "]. ";
$nombre_archivo = "status.info";
if($archivo = fopen($nombre_archivo, "w"))
{
    fwrite($archivo, $info);
    fclose($archivo);
}

$nombre_archivo = "historial.info";
if($archivo = fopen($nombre_archivo, "a"))
{
    fwrite($archivo, $info);
    fclose($archivo);
}

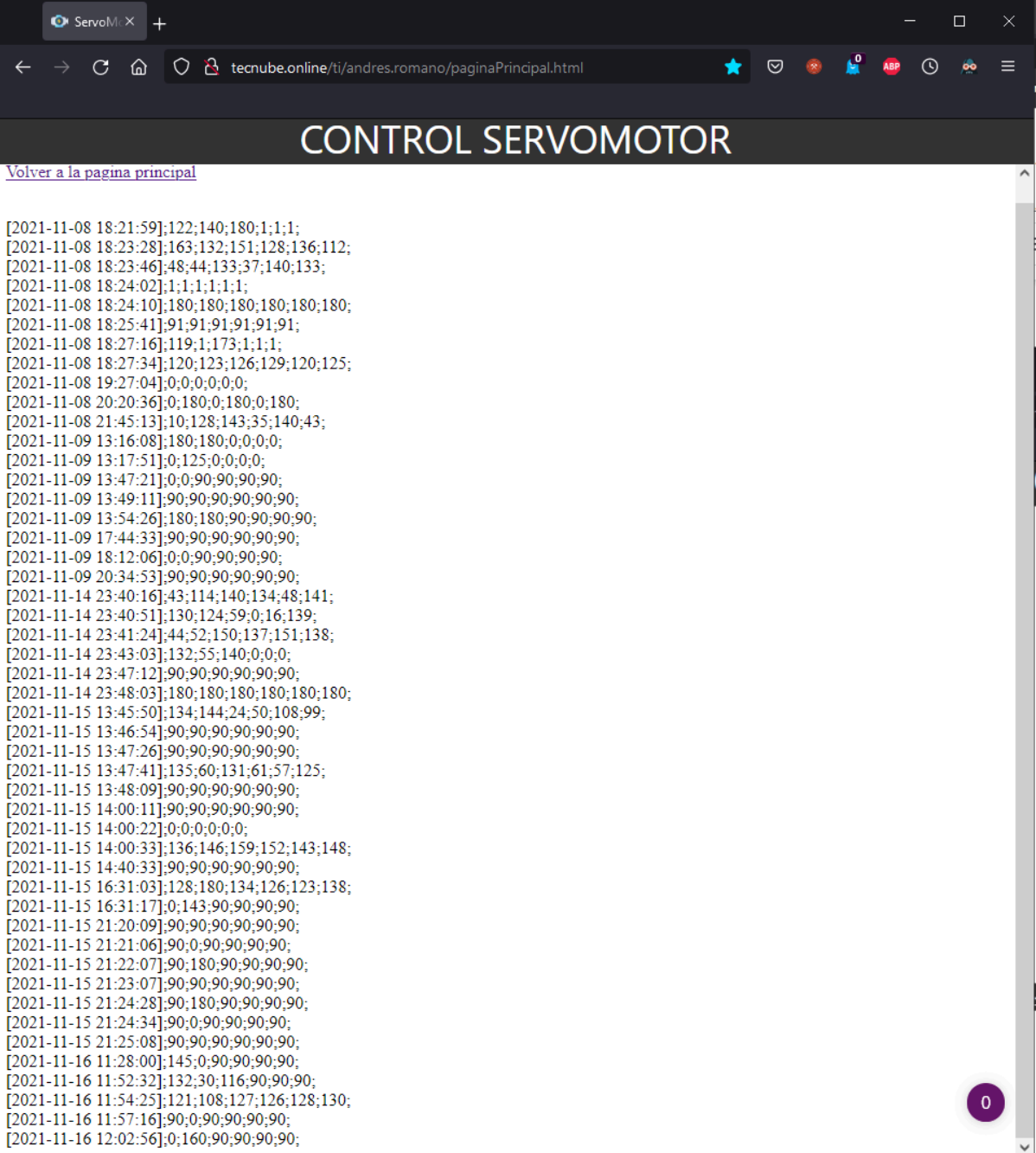
header('Location: http://tecnu.be.online/ti/andres.romano/paginaPrincipal.html');
?>

```

Finalmente se puede acceder a la página mediante el enlace:  
<http://tecnu.be.online/ti/andres.romano/paginaPrincipal.html>



Se pueden ver todos los registros guardados en **historial.info**.



ServoM X +

tecnube.online/ti/andres.romano/paginaPrincipal.html

## CONTROL SERVOMOTOR

[Volver a la pagina principal](#)

[2021-11-08 18:21:59];122;140;180;1;1;1;  
 [2021-11-08 18:23:28];163;132;151;128;136;112;  
 [2021-11-08 18:23:46];48;44;133;37;140;133;  
 [2021-11-08 18:24:02];1;1;1;1;1;1;  
 [2021-11-08 18:24:10];180;180;180;180;180;180;  
 [2021-11-08 18:25:41];91;91;91;91;91;91;  
 [2021-11-08 18:27:16];119;1;173;1;1;1;  
 [2021-11-08 18:27:34];120;123;126;129;120;125;  
 [2021-11-08 19:27:04];0;0;0;0;0;0;  
 [2021-11-08 20:20:36];0;180;0;180;0;180;  
 [2021-11-08 21:45:13];10;128;143;35;140;43;  
 [2021-11-09 13:16:08];180;180;0;0;0;0;  
 [2021-11-09 13:17:51];0;125;0;0;0;0;  
 [2021-11-09 13:47:21];0;0;90;90;90;90;  
 [2021-11-09 13:49:11];90;90;90;90;90;90;  
 [2021-11-09 13:54:26];180;180;90;90;90;90;  
 [2021-11-09 17:44:33];90;90;90;90;90;90;  
 [2021-11-09 18:12:06];0;0;90;90;90;90;  
 [2021-11-09 20:34:53];90;90;90;90;90;90;  
 [2021-11-14 23:40:16];43;114;140;134;48;141;  
 [2021-11-14 23:40:51];130;124;59;0;16;139;  
 [2021-11-14 23:41:24];44;52;150;137;151;138;  
 [2021-11-14 23:43:03];132;55;140;0;0;0;  
 [2021-11-14 23:47:12];90;90;90;90;90;90;  
 [2021-11-14 23:48:03];180;180;180;180;180;180;  
 [2021-11-15 13:45:50];134;144;24;50;108;99;  
 [2021-11-15 13:46:54];90;90;90;90;90;90;  
 [2021-11-15 13:47:26];90;90;90;90;90;90;  
 [2021-11-15 13:47:41];135;60;131;61;57;125;  
 [2021-11-15 13:48:09];90;90;90;90;90;90;  
 [2021-11-15 14:00:11];90;90;90;90;90;90;  
 [2021-11-15 14:00:22];0;0;0;0;0;0;  
 [2021-11-15 14:00:33];136;146;159;152;143;148;  
 [2021-11-15 14:40:33];90;90;90;90;90;90;  
 [2021-11-15 16:31:03];128;180;134;126;123;138;  
 [2021-11-15 16:31:17];0;143;90;90;90;90;  
 [2021-11-15 21:20:09];90;90;90;90;90;90;  
 [2021-11-15 21:21:06];90;0;90;90;90;90;  
 [2021-11-15 21:22:07];90;180;90;90;90;90;  
 [2021-11-15 21:23:07];90;90;90;90;90;90;  
 [2021-11-15 21:24:28];90;180;90;90;90;90;  
 [2021-11-15 21:24:34];90;0;90;90;90;90;  
 [2021-11-15 21:25:08];90;90;90;90;90;90;  
 [2021-11-16 11:28:00];145;0;90;90;90;90;  
 [2021-11-16 11:52:32];132;30;116;90;90;90;  
 [2021-11-16 11:54:25];121;108;127;126;128;130;  
 [2021-11-16 11:57:16];90;0;90;90;90;90;  
 [2021-11-16 12:02:56];0;160;90;90;90;90;

0

Para resolver el funcionamiento del arduino se utilizó como base el archivo **TI-ISC\_Laboratorio\_4.ino** pero con ligeros cambios.

- En primer lugar se agregó la biblioteca ESP8266HTTPClient.h
- Luego se cambiaron los parámetros para que se conecte a la red **“Abierta”** de la institución y se usó la variable **“url”** para acceder al contenido de **status.info** mediante una petición HTTP.

```
servoTecnube
//Eugenio Perdomo, Germán Torres, Andrés Romano
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <Servo.h>

#define MAX_ITEMS 7 //fecha + 6 servos
#define WAIT_TIME 20000

//-----

// Sustituir por los datos de vuestro WiFi
const char* ssid = "Abierta"; // Rellena con el nombre de tu red WiFi
const char* password = ""; // Rellena con la contraseña de la red WiFi
String url = "http://tecnube.online/ti/andres.romano/status.info";//URL host

HTTPClient http;
WiFiClient client;

String line, d_info;
Servo servo1, servo2, servo3, servo4, servo5, servo6;
int servoPosIni;
```

En **Void Setup()** se declara el monitor serial, se asignan pines a cada servo, se establece la conexión a internet y por último la posición inicial de que deben adoptar los servomotores.

```
void setup()
{
  Serial.begin(115200);

  servo1.attach(D3, 500, 2400);
  servo2.attach(D4, 500, 2400);
  servo3.attach(D5, 500, 2400);
  servo4.attach(D6, 500, 2400);
  servo5.attach(D7, 500, 2400);
  servo6.attach(D8, 500, 2400);
  delay(10);

  // Conectar a la red WiFi
  /* Configura el ESP8266 como cliente WiFi. Si no se hace
     se configura como cliente y punto de acceso al mismo tiempo */
  WiFi.mode(WIFI_STA); // Modo cliente WiFi
  WiFi.begin(ssid, password);

  // Espera a estar conectado a la red WiFi
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println();
  Serial.print("Conectandose a: ");
  Serial.println(ssid);
  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.println("Direccion IP: ");
  Serial.println(WiFi.localIP()); // Mostrar la direccion IP
  servoPosIni = 90;
  servo1.write(servoPosIni);
  servo2.write(servoPosIni);
  servo3.write(servoPosIni);
  servo4.write(servoPosIni);
  servo5.write(servoPosIni);
  servo6.write(servoPosIni);
}
```



La función **getValue()** permanece intacta con respecto a la utilizada anteriormente.

```

//*****
String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = {0, 0};
    int maxIndex = data.length()-1;

    for(int i=0; i<=maxIndex && found<=index; i++){
        if( (data.charAt(i)==separator) || (i==maxIndex) ){
            found++;
            strIndex[0] = strIndex[1]+1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }

    return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
}
//*****

```

En **Void loop()** se cambió ligeramente el formato de la petición, primero se realiza la petición y se guarda el contenido en una variable de tipo string, posteriormente mostrando el código de la petición y el contenido del **String payload**.

```

void loop() {
    Serial.print("conectando a ");
    Serial.println(url);
    http.begin(client, url); // <1KB payload

    int httpCode = http.GET(); // Realizar petición
    String payload = http.getString(); // Obtener respuesta

    Serial.printf("[HTTP] GET... code: %d\n", httpCode); //Print HTTP return code
    Serial.println("Mostrando el contenido de Status.info: ");
    Serial.println(payload); // Mostrar respuesta por serial

    http.end(); //Close connection

    // Consultar la memoria libre
    Serial.printf("\nMemoria libre en el ESP8266: %d Bytes\n\n",ESP.getFreeHeap());
    Serial.println("-----");
}

```

Para procesar la información se pasó la variable “**payload**” al llamar a la función **getValue()**.

En la condición “if” que corresponde a cada servo se agregó un delay de 3s para controlar mejor el movimiento del brazo y evitar accidentes.

```
// Procesar la info recibida
for (int i=0; i < MAX_ITEMS; i++){
    d_info = getValue(payload, ',', i);

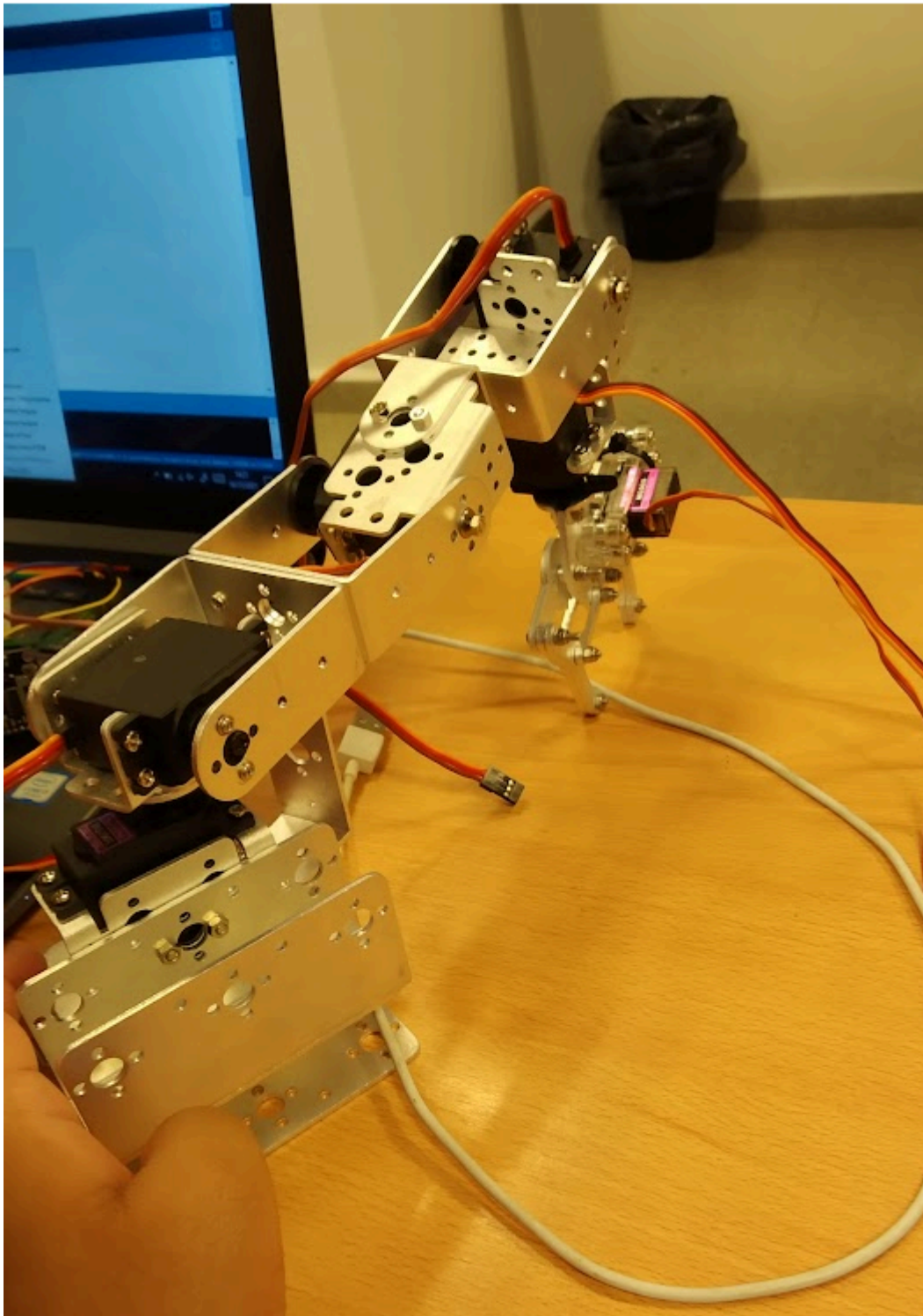
    if (i==0) {
        Serial.print("Fecha y hora: "); Serial.println(d_info);
    }
    if (i==1) {
        Serial.print("Servo"); Serial.print(i); Serial.print(" = "); Serial.println(d_info.toInt());
        servo1.write(d_info.toInt());
        delay(3000);
    }
    if (i==2) {
        Serial.print("Servo"); Serial.print(i); Serial.print(" = "); Serial.println(d_info.toInt());
        servo2.write(d_info.toInt());
        delay(3000);
    }
    if (i==3) {
        Serial.print("Servo"); Serial.print(i); Serial.print(" = "); Serial.println(d_info.toInt());
        servo3.write(d_info.toInt());
        delay(3000);
    }
    if (i==4) {
        Serial.print("Servo"); Serial.print(i); Serial.print(" = "); Serial.println(d_info.toInt());
        servo4.write(d_info.toInt());
        delay(3000);
    }
    if (i==5) {
        Serial.print("Servo"); Serial.print(i); Serial.print(" = "); Serial.println(d_info.toInt());
        servo5.write(d_info.toInt());
        delay(3000);
    }
    if (i==6) {
        Serial.print("Servo"); Serial.print(i); Serial.print(" = "); Serial.println(d_info.toInt());
        servo6.write(d_info.toInt());
        delay(3000);
    }
}
delay(WAIT_TIME);
}
```

Resultado del monitor serial:

```
.....  
Conectandose a: Abierta  
  
WiFi conectado  
Direccion IP:  
192.168.1.10  
conectando a http://tecnube.online/ti/andres.romano/status.info  
[HTTP] GET... code: 200  
Mostrando el contenido de Status.info:  
[2021-11-16 12:02:56];0;160;90;90;90;90;  
  
Memoria libre en el ESP8266: 50072 Bytes  
  
-----  
Fecha y hora: 2021-11-16 12:02:56]  
Servo1 = 0  
Servo2 = 160  
Servo3 = 90  
Servo4 = 90  
Servo5 = 90  
Servo6 = 90
```

☒ Autoscroll ☐ Mostrar marca temporal  115200 baudio

Estado del brazo previo a su montaje con la placa.



Estado del brazo mecánico luego de varias pruebas:  
En este punto el ángulo de cada servomotor era cercano a  $90^\circ$ .

