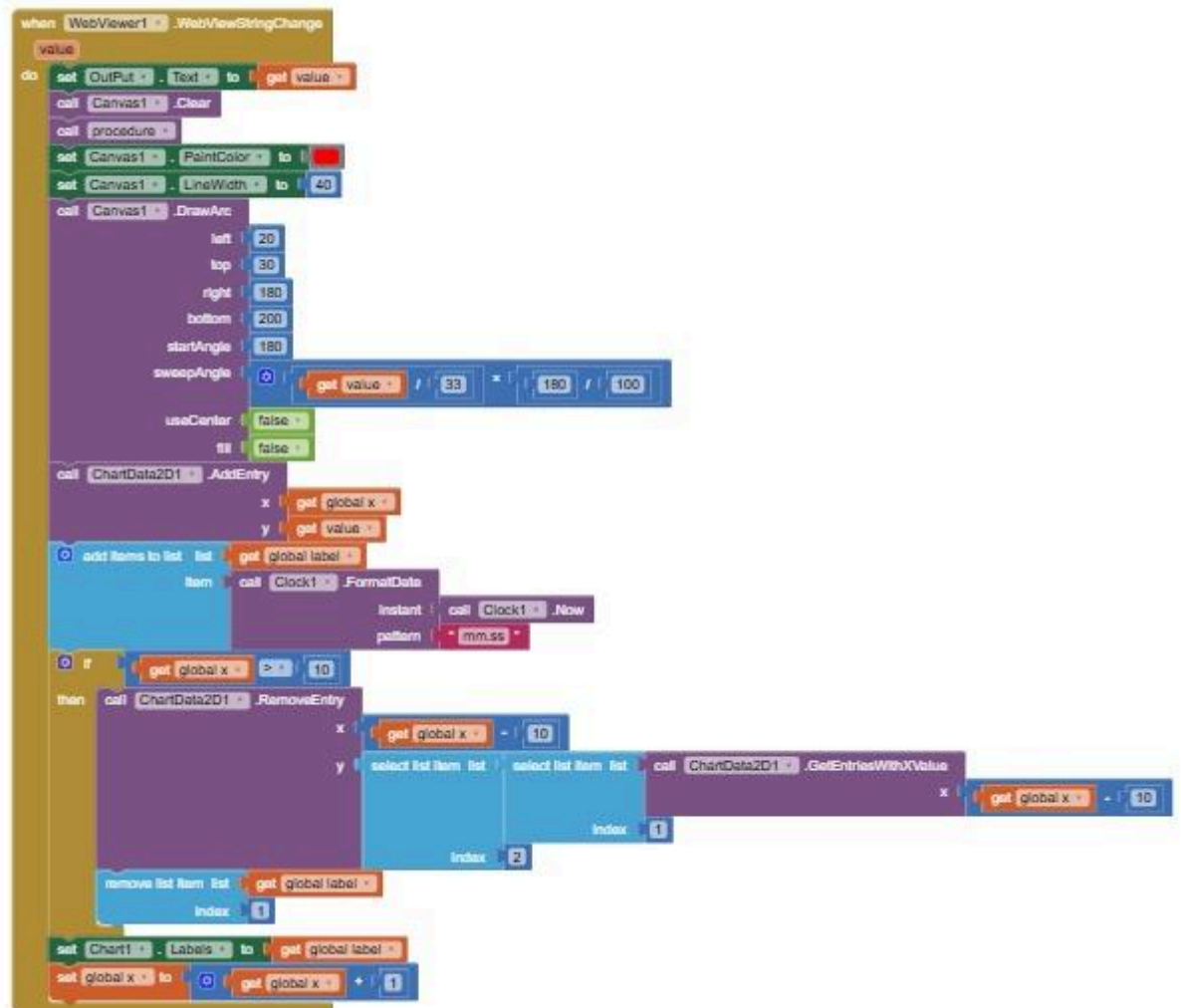


Taller conectividad-Entregable

- Video:

https://drive.google.com/drive/folders/1kQbGcQggX_dJ9MZODlcUFFMYbY-TXhCq?usp=sharing

- Capturas de pantalla:



initialize global **label** to **create empty list**

initialize global **x** to **1**

when **Conect** .Click
do
 call **WebView1** .GoToUrl
 url **TextBox1** .Text
 call **TextBox1** .HideKeyboard

to procedure
do
 set **Canvas1** .PaintColor to
 set **Canvas1** .LineWidth to **20**
 call **Canvas1** .DrawArc
 left **20**
 top **30**
 right **180**
 bottom **200**
 startAngle **180**
 sweepAngle **180**
 useCenter **false**
 fill **false**

- Código .ino Comentado

1. `#include <WiFiManager.h>` // Librería para gestionar conexiones WiFi con portal cautivo
2. `//#include <WiFiClient.h>` // Comentado, no se usa directamente
3. `#include <WebServer.h>` // Servidor web HTTP para ESP8266/ESP32
4. `#include "index.h"` // Archivo que contiene el HTML de la página principal
- 5.

```

6. WebServer server(80);      // Crea un servidor web en el
    puerto 80
7.

8. // Función para manejar la raíz del servidor ("/")
9. void handleRoot() {
10.     String s = MAIN_page; // MAIN_page es una cadena HTML
        definida en "index.h"
11.     server.send(200, "text/html", s); // Envía la página
        HTML al navegador
12. }
13.
14. // Función para manejar la ruta "/adc" y devolver el
    valor leído del pin analógico
15. void handleADC() {
16.     int a = 0;
17.     a = analogRead(A0); // Lee el pin analógico A0
        (conectado a un potenciómetro)
18.     a = map(a, 0, 4095, 0, 3300); // Convierte el valor a
        milivoltios (0-3300 mV)
19.     String adcValue = String(a); // Convierte el valor
        numérico a texto
20.     server.send(200, "text/plane", adcValue); // Envía el
        valor como texto plano al cliente
21. }
22.
23. void setup() {
24.     // Inicializa el puerto serie a 115200 baudios para
        depuración
25.     Serial.begin(115200);
26.
27.     // Crea un objeto WiFiManager para gestionar la conexión
        WiFi
28.     WiFiManager wm;
29.
30.     // Puedes descomentar la siguiente línea para borrar
        credenciales WiFi previas
31.     // wm.resetSettings();
32.
33.     // Intenta conectarse automáticamente a WiFi con SSID y
        contraseña dados
34.     // Si falla, crea un portal cautivo con ese SSID y
        contraseña para configuración manual
35.     bool res = wm.autoConnect("ConnectAP", "1234567890");
36.
37.     if (!res) {
38.         Serial.println("Fallo la conexion"); // Si falla la
            conexión
39.         // Aquí podrías reiniciar el dispositivo o intentar otra
            acción
40.     } else {

```

```
41.     Serial.println("Conectado :"); // Éxito al conectar
42. }
43.
44. // Asocia las rutas del servidor con las funciones
    correspondientes
45.     server.on("/", handleRoot); // Ruta principal
46.     server.on("/adc", handleADC); // Ruta de lectura
        analógica
47.
48.     server.begin(); // Inicia el servidor web
49. }
50.
```