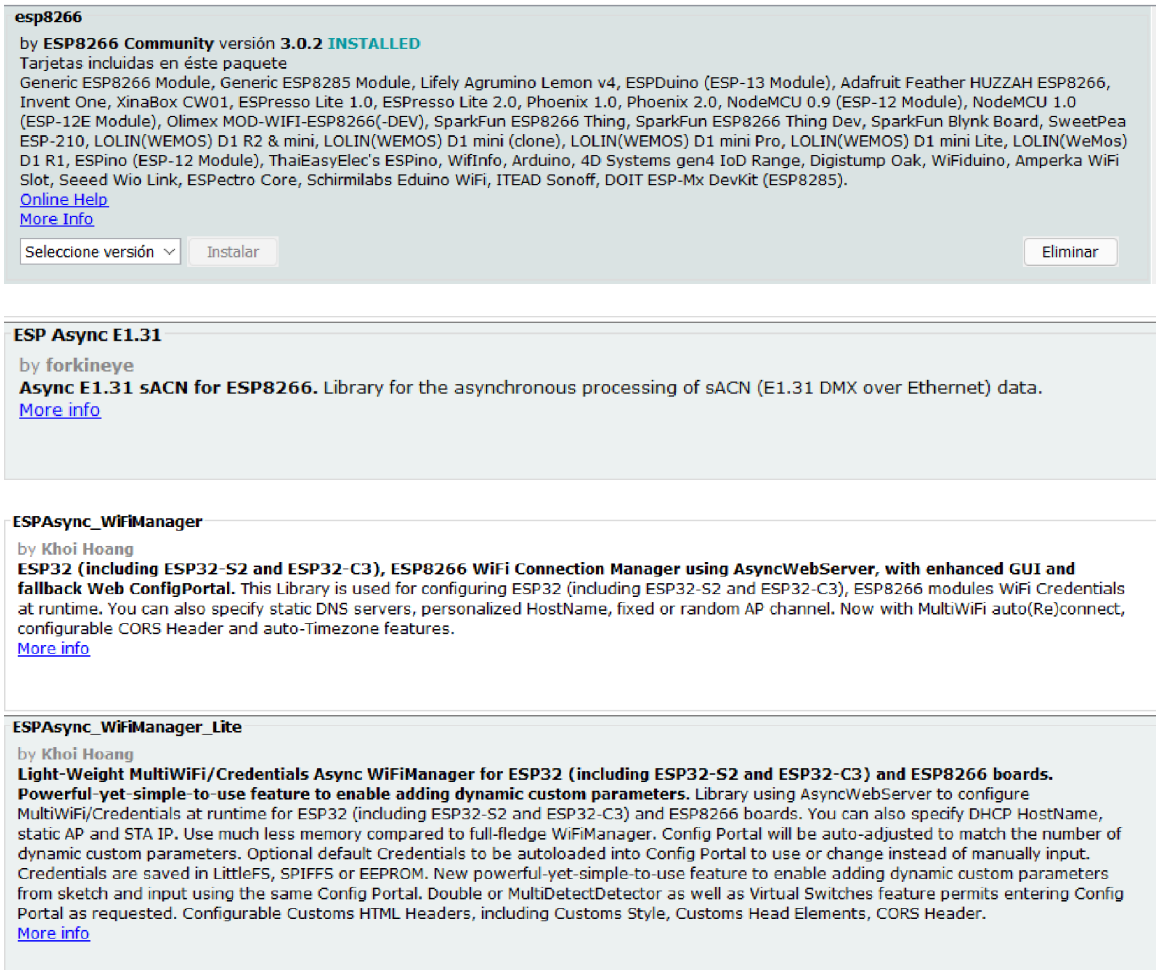
Librerías instaladas



Definición de librerías

```
#include <ESP8266WiFi.h>
```

```
#include <ESPAsyncTCP.h>
```

```
#include <ESPAsyncWebServer.h>
```

Código

```
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>

const char* ssid = "Nombre de la red";
const char* password = "Contraseña de la red";

const int led_pin = 4;
String slider_value = "0";

const char* input_parameter = "value";

AsyncWebServer server(80);

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <title>ESP8266 Control De Brillo Servidor Web</title>
```

```
  <style>
    html {font-family: Times New Roman; display: inline-block; text-align: center;}
    h2 {font-size: 2.3rem;}
    p {font-size: 2.0rem;}
    body {max-width: 450px; margin:0px auto; padding-bottom: 25px;}
    .slider { -webkit-appearance: none; margin: 14px; width: 360px; height: 25px; background: #FF0000;
      outline: none; -webkit-transition: .2s; transition: opacity .2s;}
    .slider::-webkit-slider-thumb {-webkit-appearance: none; appearance: none; width: 35px; height: 35px; background:#01070a; cursor: pointer;}
    .slider::-moz-range-thumb { width: 35px; height: 35px; background: #01070a; cursor: pointer; }
  </style>
</head>
<body>
  <h2>ESP8266 Control De Brillo Servidor Web</h2>
  <p><span id="textslider_value">%SLIDERVALUE%</span></p>
```

```
  <p><input type="range" onchange="updateSliderPWM(this)" id="pwmSlider" min="0" max="255" value="%SLIDERVALUE%" step="1" class="slider"></p>

<script>

function updateSliderPWM(element) {

  var slider_value = document.getElementById("pwmSlider").value;

  document.getElementById("textslider_value").innerHTML = slider_value;

  console.log(slider_value);

  var xhr = new XMLHttpRequest();

  xhr.open("GET", "/slider?value="+slider_value, true);

  xhr.send();

}

</script>

</body>

</html>

)rawliteral";


String processor(const String& var){

  if (var == "SLIDERVALUE"){
```

```
    return slider_value;
  }
  return String();
}

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

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting...");
  }

  Serial.println(WiFi.localIP());

  server.on("/", HTTP_GET, [](AsyncWebServerRequest
*request){
    request->send_P(200, "text/html", index_html, processor);
  });
```

```
  server.on("/slider", HTTP_GET, [] (AsyncWebServerRequest
*request) {
    String message;
    if (request->hasParam(input_parameter)) {
      message = request->getParam(input_parameter)->value();
      slider_value = message;
      analogWrite(led_pin,slider_value.toInt());
    }
    else {
      message = "No message sent";
    }
    Serial.println(message);
    request->send(200, "text/plain", "OK");
  });

  server.begin();
}

void loop() {
```

```
}
```