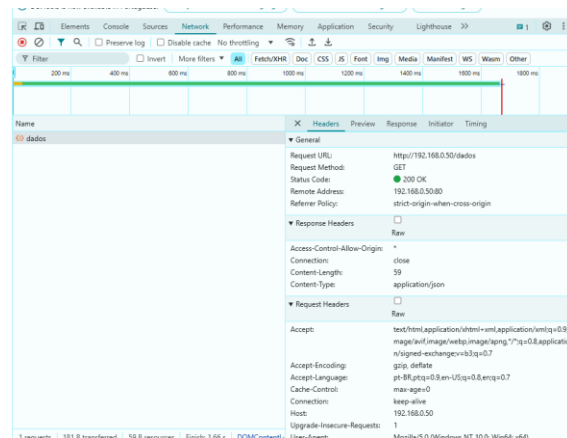


## Servidor ESP32.

("unidade":3792, "solo":"Solo Encharcado!", "segundos":142)



## Trocando arquivos.

Por meio do código, O servidor entrega por JSON as informações de: • valor do sensor • Função de verificação do estado do solo • Segundos deis da ultima vez que a bomba d'agua foi ligada. Que estão sendo entregues para o Servidor com o dominio IP/dados. (192.168.0.50/dados)

## Código

```
#include <WiFi.h>
```

```
#include <WebServer.h>
```

```
// definir senha e nome do wifi que o ESP32 vai se conectar.
```

```
const char* ssid = "*****";
```

```
const char* password = "*****";
```

```
WebServer server(80); // cria um servidor na porta 80
```

```
// conectar ao Wi-Fi
```

```
WiFi.begin(ssid, password);
```

```
//diz que está conectando no serial e enquanto não estiver em situação de conectado, ele escreverá pontos até conectar.
```

```
Serial.print("Conectando-se ao WiFi...");
```

```
while (WiFi.status() != WL_CONNECTED) {
```

```
    delay(1000);
```

```
    Serial.print(".");
```

```

    }

    //quand estiver conectado, pula uma linha, e escreve o IP do Servidor criado
    pelo ESP32 que será usado depois

    Serial.println("\nConectado ao WiFi!");

    Serial.print("Endereço IP: ");

    Serial.println(WiFi.localIP());


    // configura um servidor com o nome ip/dados para envio de informações.

    server.on("/dados", []() {

//configuração de entrega de informações para o servidor

        valor = analogRead(Sensor); // Atualiza o valor do sensor

        String umidade = String(valor);

        String solo = Solo(Sensor);


        //cria uma string escrita da maneira correta em que um arquivo .JASON é
        escrito.

        String resultado = "{\"umidade\":\"" + umidade + "\", \"solo\":\"" + solo + "\",
        \"segundos\":\"" + String(segundos) + "\"}";

        // utiliza o protocolo de permissão de troca de informações entre site
        chamado Cross-Origin Resource Sharing (CORS)

        // ele permite que o site html possa puxar informações do site criado pelo esp,
        Como foi colocado um "*" no parametro, ele permitira que qualquer um possa
        puxar as informações do ESP32.

        server.setHeader("Access-Control-Allow-Origin", "*");

        /*aqui ele enviará para o servidor com um código de confirmação 200, que para
        o HTML significa OK. Caso de erro de retorno do sinal ele retornara um 404,
        oque não é um ok.

        tambem envia um arquivo chamado "application/json" com as informações de
        resultado para o servidor, tal arquivo será no futuro puxado pelo HTML.  */

        server.send(200, "application/json", resultado);

    });

```

```
// inicia o servidor

server.begin();

}
```

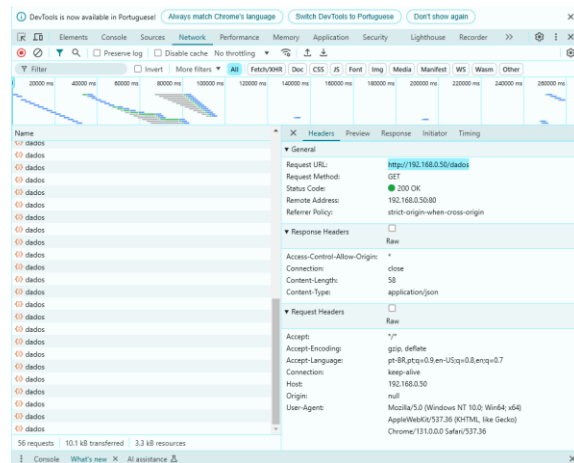
## Server Cliente Atualizando.

### Monitor de Umidade do Solo

Umidade do Solo: 3778

Tempo desde a última vez ligada a bomba: 328 Segundos.

Condição atual do Solo: Solo Encharcado!



O Servidor Cliente <http://127.0.0.1:5500/> esta por meio das configurações do código, requisitando arquivos do Servidor ESP32 e retornando um Sucesso sempre que a requisição deu certo.

O servidor está requerendo as informações do JSON:

- Valores do sensor.
- Estado do solo
- Segundos

Também, mostrando na Interface do Cliente as informações separadamente obtidas do Servidor do ESP32.

```
<!DOCTYPE html>
```

```
<html lang="pt">
```

```
// aqui fica oque é importante para o código e que não aparece na pagina para o cliente.
```

```
<head>
```

```
//configuração do texto da página
```

```

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

//título da página

<title>Monitor de Umidade do Solo</title>

//script para a obtenção de informações do esp32 por meio de conexão peer-
to-peer

<script>

    const ip = '192.168.0.50'; // IP do ESP ps: não me hackeiem >:(

//função para obter o arquivo.json enviado para o servidor do ESP32

    function obterDados() {

//aqui é a parte importante. aqui ele vai buscar pelo fetch no site http:// o IP do
esp/dados. Site este que está sendo criado pelo ESP32.

        fetch(`http://${ip}/dados`)

            .then(response => response.json()) // faz a resposta como JSON

            .then(data => {

                //entrega informações para algum lugar do código por Id.

                document.getElementById("umidade").innerText = "Umidade do
Solo: " + data.umidade;

                document.getElementById("Solo").innerText = "Condição atual do
Solo: " + data.solo;

                document.getElementById("Segundos").innerText = "Tempo desde
a última vez ligada a bomba: " + data.segundos + " Segundos.";

            })

            //caso o fetch tenha entregue um erro, ele escreverá no Console do
site um erro.

            .catch(error => {

                console.error('Erro:', error);

            });

    }

```

```
// atualiza os dados a cada 2 segundos as informações do site
setInterval(obterDados, 2000);
</script>
</head>
//aqui é oque vai aparecer no site
<body>
    <h1>Monitor de Umidade do Solo</h1>
    //coloca um id para cada linha para poder receber informaçõesa pela head
    <p id="umidade">Umidade do Solo: Carregando...</p>
    <p id="Segundos">Segundos deis da última vez:</p>
    <p id="Solo">Condição Atual do Solo:</p>
</body>
</html>
```