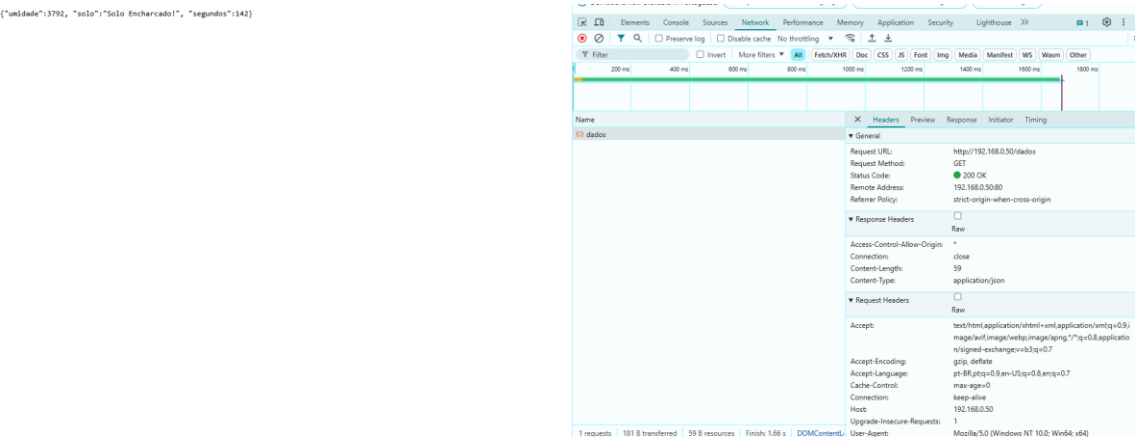


Servidores e sua performance.

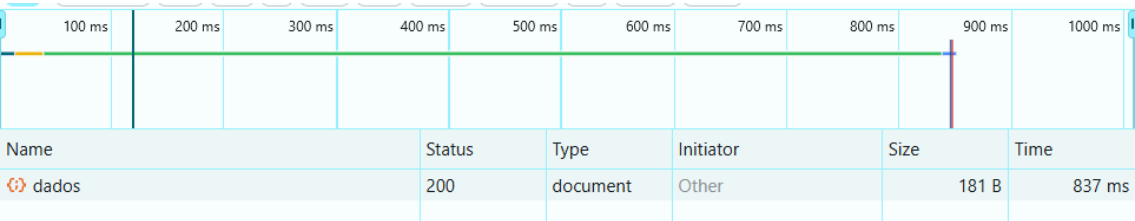
Servidor ESP32.



O ESP32 cria um servidor com o IP/dados da porta 80 que é uma porta default do HTML, usando a rede WiFi previamente conectada no código.

A partir disto o ESP32 começara a enviar as informações do Sensor e outras funções em formato JSON para o server buscando um OK como resposta do Servidor ao ESP em caso de sucesso.

Performance.

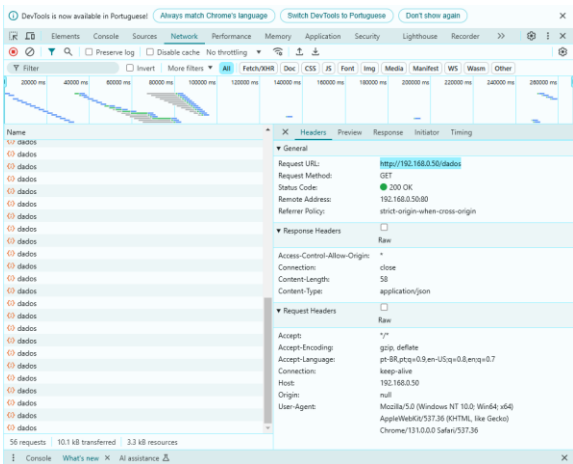


O servidor está recebendo as informações com um delay abaixo de 1 SEG o que não causa nenhum tipo de problema para leitura futuras por meio de outros servidor, pois, está definido para ele atualizar a cada 1 seg o servidor.

# Servidor Cliente

## 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!



Servidor cliente feito com HTML que busca as informações em JSON do Domínio IP/dados que é o servidor criado pelo ESP32. Com sucesso, o Servidor Cliente atualiza a cada 2 Segundos, buscando novas informações do Servidor do ESP32.

## Performance HTML

Name	Status	Type	Initiator	Size	Time
dados	200	fetch	site HTML JSON JAVA SC	181 B	277 ms
dados	200	fetch	site HTML JSON JAVA SC	181 B	288 ms
dados	200	fetch	site HTML JSON JAVA SC	181 B	295 ms
dados	200	fetch	site HTML JSON JAVA SC	181 B	299 ms
dados	200	fetch	site HTML JSON JAVA SC	181 B	311 ms
dados	200	fetch	site HTML JSON JAVA SC	181 B	308 ms
dados	200	fetch	site HTML JSON JAVA SC	181 B	218 ms
dados	200	fetch	site HTML JSON JAVA SC	181 B	328 ms
dados	200	fetch	site HTML JSON JAVA SC	181 B	335 ms
dados	200	fetch	site HTML JSON JAVA SC	181 B	343 ms

O delay da entrega das informações não altera ou prejudica o entendimento do cliente, pois, a atualização aparece no tempo correto do requerimento do servidor html para o servidor ESP32. O delay do Servidor ESP32 + O delay do Servidor Cliente não supera a taxa de atualização os servidores, o que mantém os servidores fluidos e sem erros.