O comando ‘npm install’ serve para instalar o node na maquina e sincronizar eles com os packages que estão nos arquivos.

O ‘npm start’ inicializa o servidor node para que possa receber os dados e atualizar os graficos.

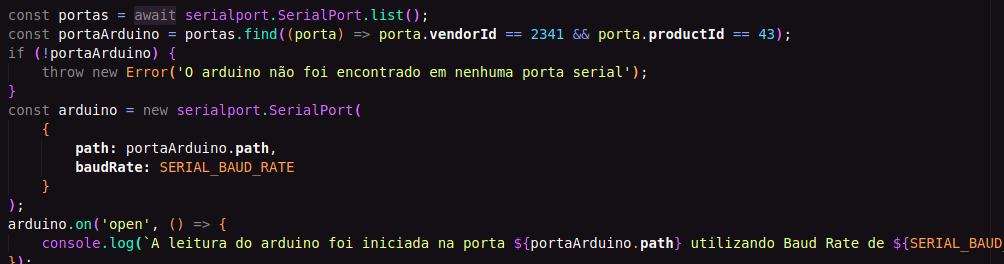
As primeiras 7 linhas do codigo main.js estão definindo os valores de portas e taxa de transferencia de dados.

O primeiro JSON está definindo uma caixa de dados que serão definidas mais pra frente no codigo.

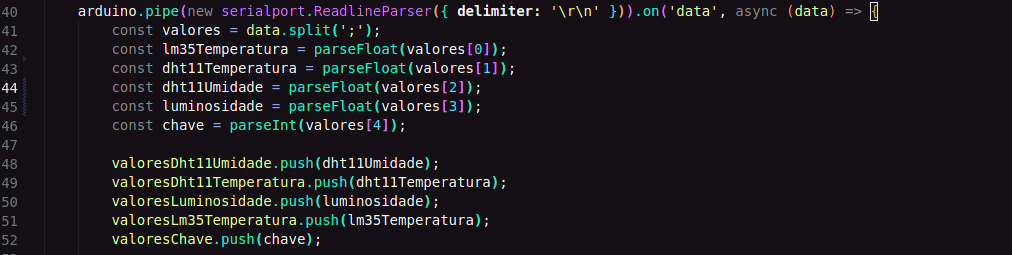
Da linha 26 até a linha 65 o codigo está primeiramente pegando a porta que está sendo identificado o arduino e se caso não for identificado nenhuma porta, mostrará um erro no console. Após identificar a porta do arduino, ele captura os dados gerados pelos sensores e armazena em variáveis. Esses valores vão ficar armazenados na JSON parse e também serão puxadas pela primeira JSON citada. Se o arduino identificar algum erro, esse erro será exibido no console.

DA linha 67 até a linha 98 está sendo configurado o servidor onde os valores que são gerados e armazenados nas JSON’s são retornados para ele.

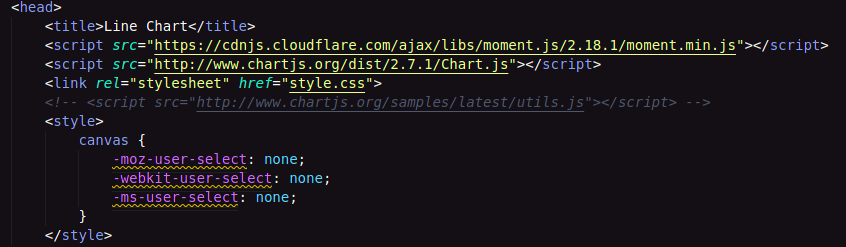
O código da API está, em sua main.js, primeiramente definindo a taxa de entrada de dados (no caso isso seria a const SERIAL\_BOUND\_RATE) e em seguida ele procura em qual local está conectado o arduino e se caso não for encontrado em nenhuma porta, dará erro.

Após localizar a porta, ele se inicia e mostra uma mensagem no console, indicando qual porta está sendo lida.

Logo após, ele pega as informações que estão sendo transmitidas pelos sensores m um vetor horizontal e separa eles por ‘; ‘ e transforma elas em um vetor vertical.



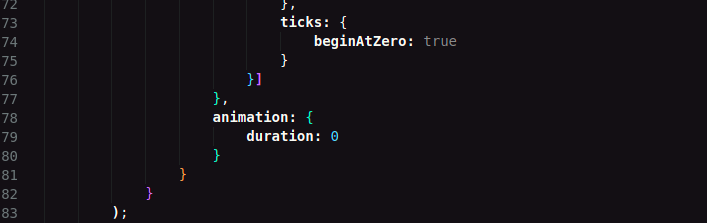
No código html, está primeirante pegando bibliotecas jChartJS para ajudar no código.



Logo após estamos criando seções onde os graficos vão ficar expostos.

E utilizando o chartJS estamos criando os gráficos no script

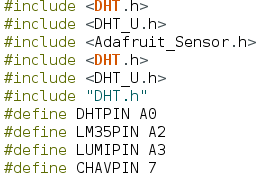


Todos os códigos do ChartJS são iguais, mudando apenas o seu nome e os dados que são inseridos neles.

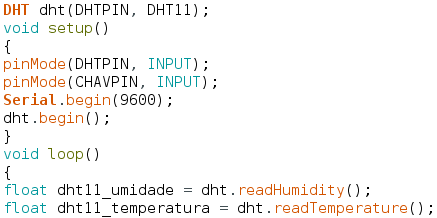
Para aparecer apenas o nosso sensor, definimos as seções dos outros gráficos como display none, assim fazendo elas sumirem da tela mas sem deletá-las do código.



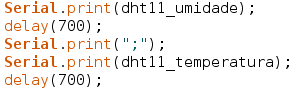
Também mudamos a ordem que nosso gráfico irá aparecer no vetor(na print do main.js já está alterado a ordem).



Nessa primeira parte do código arduino, nós incluímos os sensores, as bibliotecas e as entradas em que os sensores estão ligados.



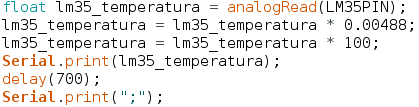
Nesta parte do código está localizado o loop em que o sensor dht11 está inserido.



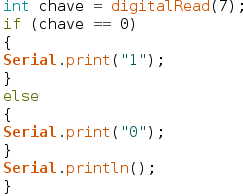
Aqui é feita a exibição dos resultados capturados dos sensores.



Aqui é exibido o resultado do sensor de luminosidade.



Nesse trecho é feito o calculo que fará com que o resultado apareça corretamente no monitor serial e logo depois a exibição é feita.



Nesse pedaço o sensor de presença é ativado caso haja algo bloqueando sua recepção deixando seu resultado como 0.