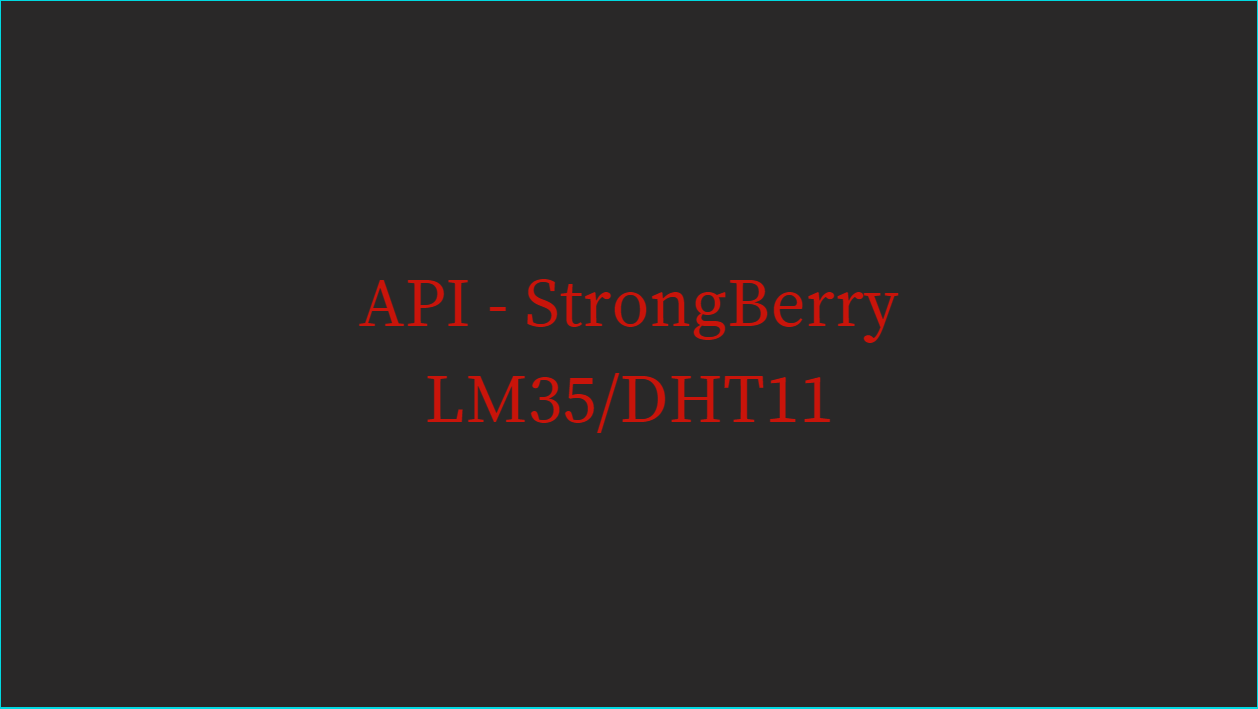
# Falas Apresentação – João Ruas

* Slide 19

Começando a Apresentação da API – “Agora vamos falar sobre o desenvolvimento de nossa API”.



* Slide 20

Transição – “Primeiro de Tudo mostraremos o funcionamento dos Gráficos da API, já com nossas especificações de temperatura e umidade”.



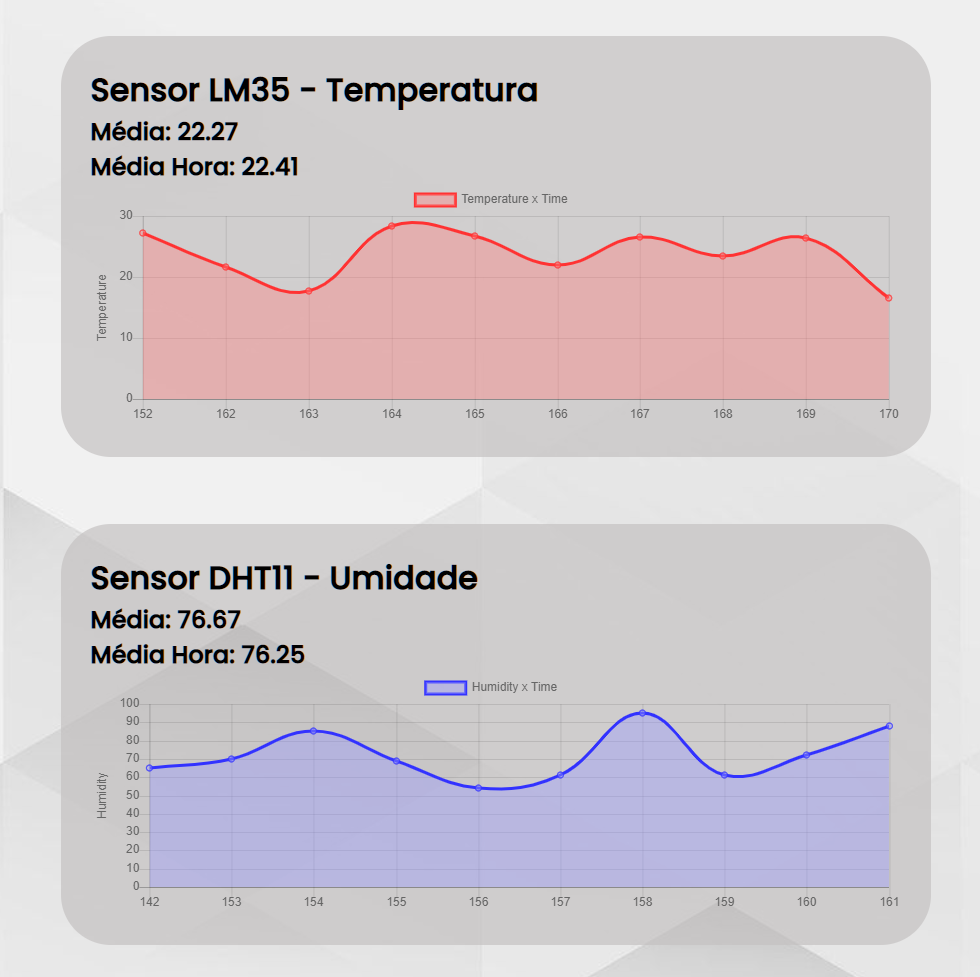
* Google Chrome

*(Observação: Quem tiver com o Slide no PC, desculpa mas por favor abre essas coisas no “Google Chrome”. Só para dar continuidade ao Slide e puxar o saco da Professora porque ela disse que fez a API baseada no padrão do Chrome ;D)*

Demonstração dos Sensores – “Depois de darmos um ‘*npm start*’, já podemos ver que conseguimos com sucesso ao alocar a API em nosso site institucional.”

“O primeiro gráfico se trata do Sensor de Temperatura LM35, com sua temperatura parametrizada para variar na faixa entre: 16.1 o mínimo e 28.9 o máximo”.

“Já o segundo gráfico pertence ao Sensor de Umidade e Temperatura DHT11. Sua temperatura está na mesma faixa que o LM35, mas a umidade está parametrizada para variar entre: 62 no mínimo e 73 no máximo ”.



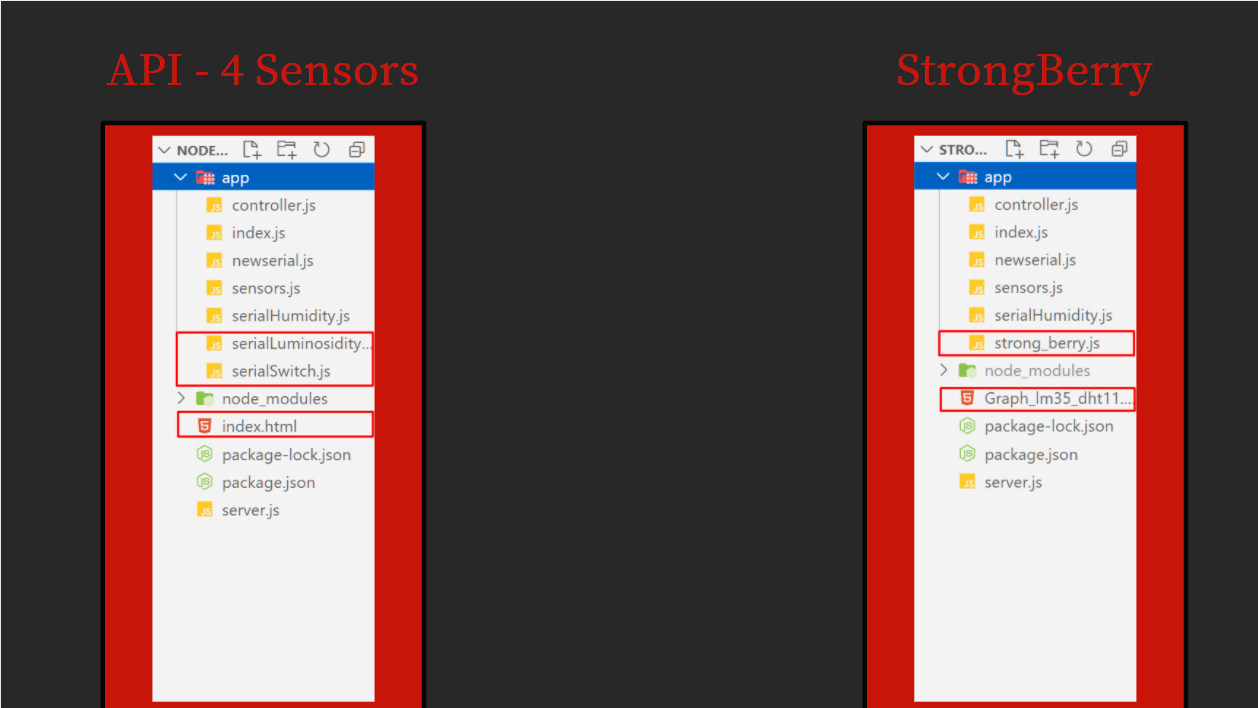
* Slide 21

Transição – “Agora irei mostrar mais especificamente aonde alteramos a API para se adaptar as necessidades de nosso projeto”.



* Slide 22

Alterações Print – “Para as Alterações mais drásticas. Podemos ver que nesse slide dois arquivos ‘*javascript*’ que são referentes aos sensores de Luminosidade e Presença foram retirados e o arquivo ‘*index.html*’ foi renomeado”.



* Slide 23

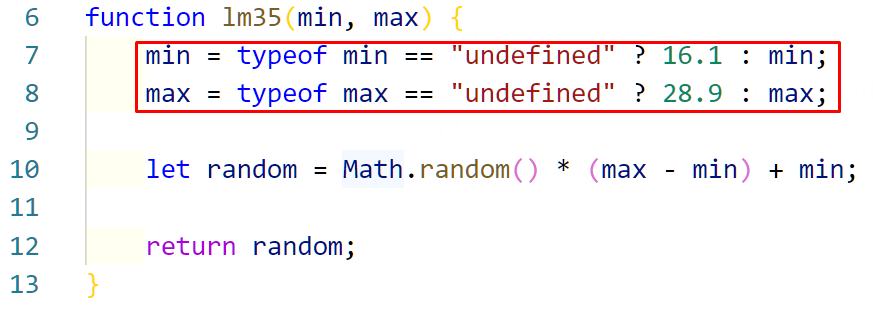
Transição para o VScode – “Bom agora irei mostrar algumas mudanças mais especificas dentro do código da API”.



* Parte 1 – VsCode

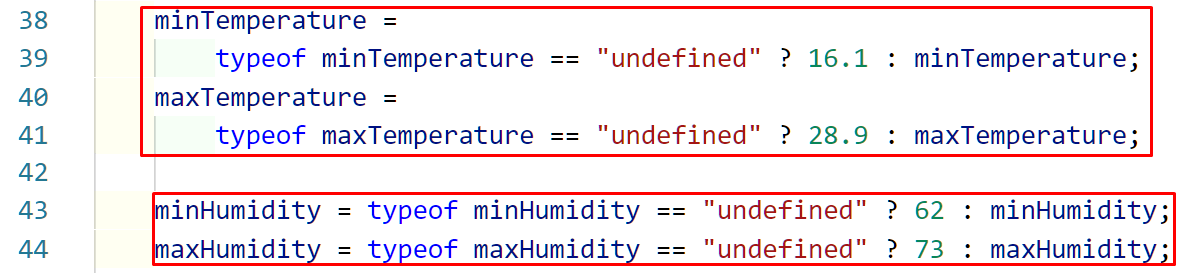
Dentro do arquivo *sensors.js* – “Primeiramente aqui dentro do arquivo *sensors.js* é o local aonde são definidos os parametros de minimo e maximo dos sensores, além de ser o local aonde são gerados os dados aleatorios”.

“Nessa *function*  chamada *lm35* é aonde definimos os parametros do Sensor de Temperatura LM35”.



Ainda dentro do arquivo *sensors.js* – “Agora um pouco mais abaixo podemos ver a *function* chamada *dht11* que também é aonde são parametrizados o mínimo e máximo de temperatura e umidade do Sensor DHT11”.

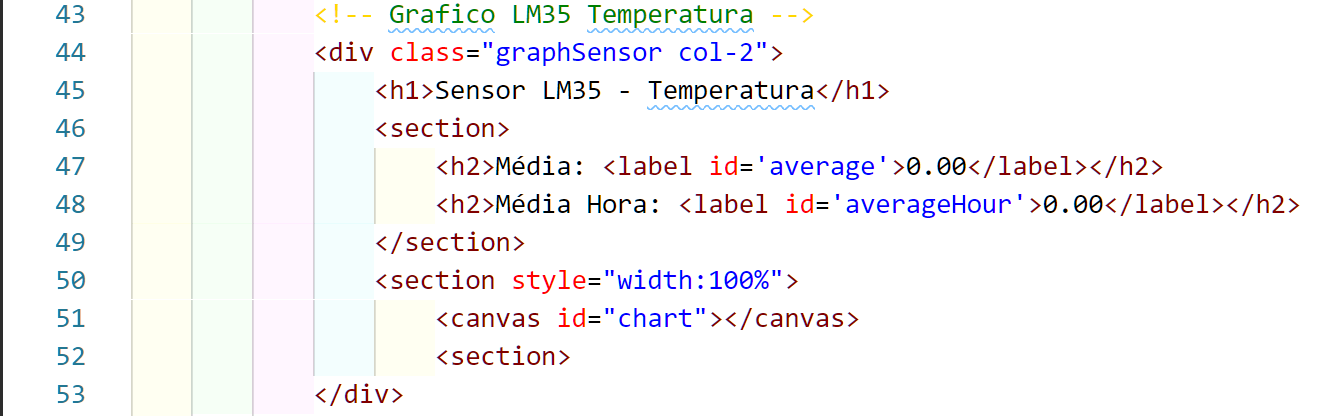




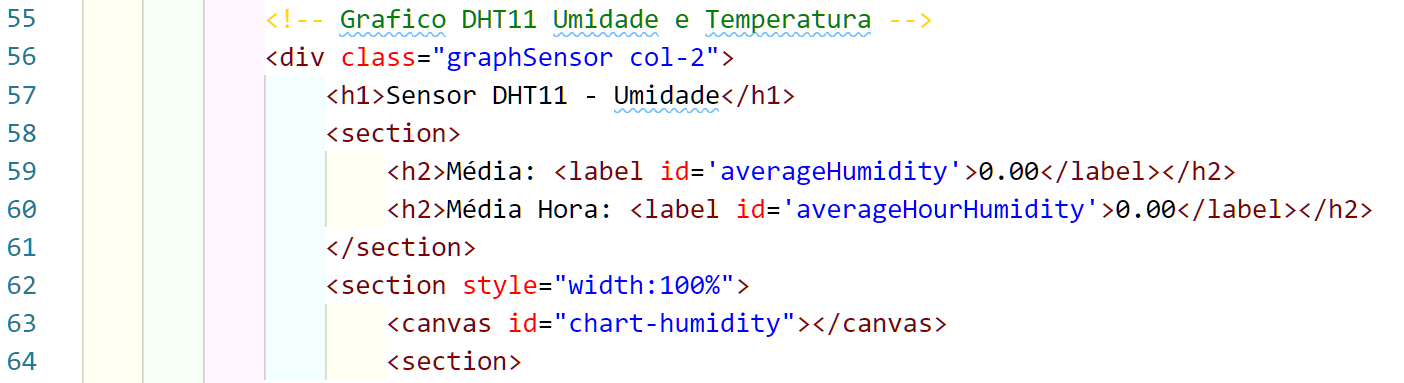
* Parte 2 – Vscode

Dentro do aquivo *Graph\_lm35\_dht11.html* – “Agora para explicação de nosso arquivo html”.

“Aqui dentro dessa *div* na linha 44 tem um *canvas* que ira receber as os dados gerados do Arduino do Sensor LM35 a partir dos arquivos *Json* do *controller.js* disponibilizados para ele a partir da porta 3000 que foi aberta pelo *server.js* e irá mostra-los em um contexto 2D”.



“Do mesmo modo na linha 56 tem uma *div* com um *canvas* dentro que ira receber os dados do Arduino do Sensor DHT11”.



* Slide 25

Modelo de Projeto – “Para finalizar essa parte fizemos um breve diagrama para demonstrar o ciclo de funcionamento da API. Aonde acreditamos ser baseado em um Padrão de Projeto chamado *MVC* que tem como objetivo uma camada que faz intermédio entre o back-end e o Client Side utilizando os métodos de *requests* e *response*, no caso essa camada intermediaria é o *controller.js* como podem ver no slide atual”.

