Após baixar o arquivo da API e instalarmos o Node.js, precisamos analisar o código para podemos adaptar.. segue alguns pontos.

Não faremos nenhuma alteração no diretório node\_modules, trabalharemos apenas com o diretório app e o arquivo index.html

No arquivo index.html temos tudo o que precisamos para encaixar 1 gráfico já rodando o sensor LM35:

* Primeiro, precisamos copiar as referências e o style do canvas.Texto

  Descrição gerada automaticamente
* Encaixamos espacialmente o gráfico em nosso arquivo (detalhe para mudança de id do ‘chart’ e da label ‘average’).

Texto

Descrição gerada automaticamente

* Agora, dentro do script, faremos as configurações visuais do gráfico, de acordo com nosso tema.

**Gráfico 1**

**Texto

Descrição gerada automaticamente**

**Gráfico 2**

**(atenção às novas variáveis, criadas conforme as anteriores.)**

**Texto

Descrição gerada automaticamente**

* Depois duplicamos, também, os atributos para dos dados processados em cada gráfico.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

* Agora, dentro da função **get\_data()**, adaptamos, também, para que haja a atualização de dados contínua em ambos os gráficos.

**Texto

Descrição gerada automaticamente**

**newserial.js**

Para que os dados sejam atualizados corretamente, temos que fazer as devidas alterações no arquivo **newseriel.js**. Divisão das leituras.

Texto

Descrição gerada automaticamente

Agora, **fake\_data()** adaptado.

Texto

Descrição gerada automaticamente

Finalizando **newserial.js** com..Tela preta com letras brancas

Descrição gerada automaticamente

**controller.js**

Para que as medias (‘average’) funcionem, atualizamos o arquivo **controller.js**

Texto

Descrição gerada automaticamente