

Lista 16 – Programação Funcional e APIs

1. Programação funcional/declarativa

a. Complemente o código abaixo, fazendo o que se pede:

```
const produtos = [
  {nome: 'Notebook Samsung', preco: 4999.99, vendido: true, tipo: 'notebook'},
  {nome: 'Notebook Sony', preco: 5999.99, vendido: false, tipo: 'notebook'},
  {nome: 'iPad', preco: 6999.99, vendido: false, tipo: 'tablet'},
  {nome: 'iPhone', preco: 7999.99, vendido: true, tipo: 'smartphone'},
  {nome: 'Copo de plástico', preco: 1.99, vendido: false, tipo: 'copo'},
  {nome: 'Copo de vidro', preco: 10.99, vendido: true, tipo: 'copo'},
];

console.table(produtos);
```

- Utilizando o método `filter()` filtre somente os produtos cujo preço é maior do que 5000 (o código irá mostrar todo o objeto).
- Encadeie com a função `map()` para mostrar somente os nomes dos produtos cujo preço seja maior do que 5000.
- Faça duas versões do código, uma colocando as *arrow functions* diretamente dentro das funções `filter()` e `map()` e outra atribuindo as *arrow functions* a variáveis e usando estas dentro das funções `filter()` e `map()`.

b. Complemente o código abaixo, fazendo o que se pede:

```
const carrinho = [
  { "nome": "Borracha", "preco": 3.45 },
  { "nome": "Caderno", "preco": 13.90 },
  { "nome": "Lápis", "preco": 2.5 },
  { "nome": "Caneta", "preco": 7.5 }
];
```

- Use o método `JSON.parse()` junto com a função `map()` para converter as *strings* do *array* acima em objetos JavaScript.
- Use a função `map()` recuperar somente os preços e atribuí-los ao *array* `precos`.
- Imprima o *array* `precos`.
- As duas primeiras opções podem ser feitas separadamente ou em uma instrução somente.

c. Use o método `reduce()` para receber uma série de 5 valores em um *array* e retornar o [produto](#) entre eles.

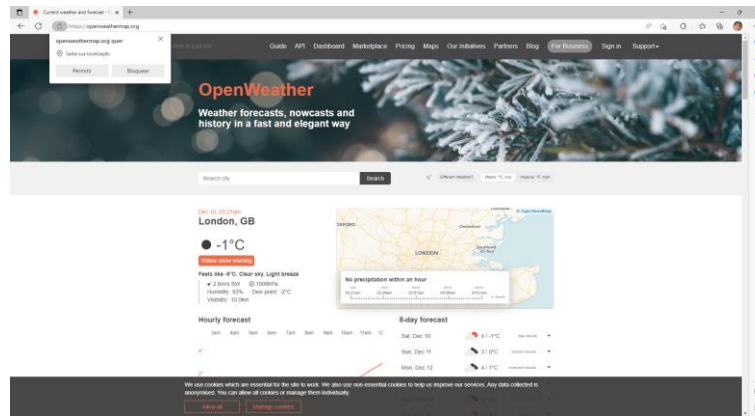
d. Complemente o código abaixo, fazendo o que se pede:

```
const alunos = [
  { nome: 'Fulana', nota: 7.3, bolsista: false },
  { nome: 'Beltrana', nota: 9.2, bolsista: true },
  { nome: 'Sicrana', nota: 9.8, bolsista: false },
  { nome: 'Maria das Dores', nota: 8.7, bolsista: true },
];
```

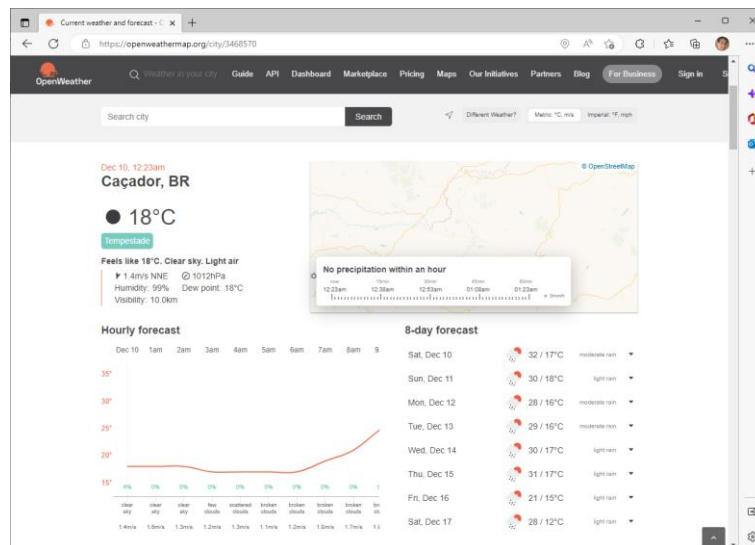
- Use as funções `map()` e `reduce()` para produzir o somatório das notas de todos os alunos e o imprima.
- Use as funções `map()` e `reduce()` para retornar `true` se TODOS os alunos forem bolsistas ou `false` se PELO MENOS UM dos alunos não for bolsista.
- Use as funções `map()` e `reduce()` para retornar `true` se PELO MENOS UM dos alunos receber bolsa ou `false` se NENHUM dos alunos for bolsista.

2. API de previsão do tempo.

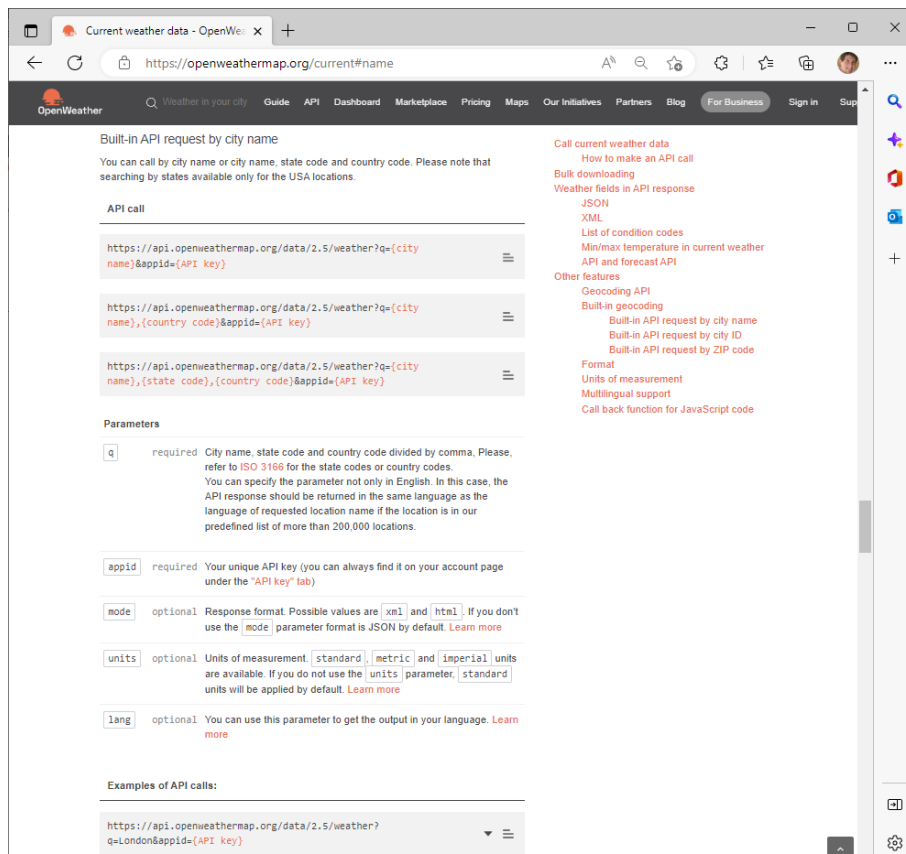
- a. Acesse o site <https://openweathermap.org/> e clique em *Permitir* ao ser solicitado pelo navegador.



- b. Faça uma consulta do tempo de sua cidade, usando o campo *Search city* como mostrado abaixo:



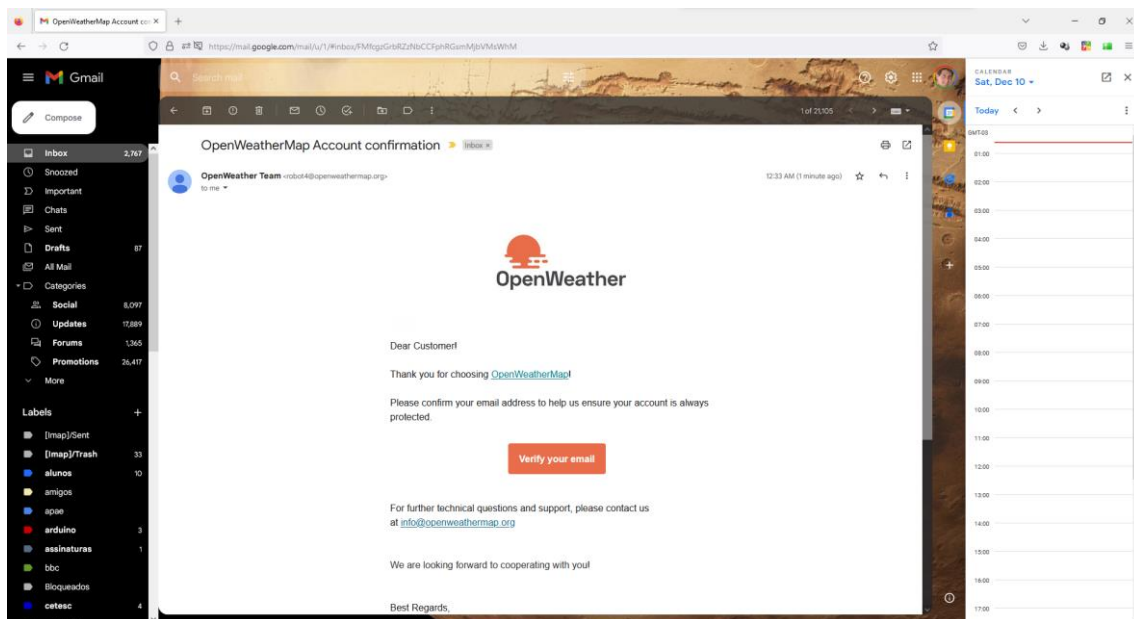
- c. A página da documento abaixo mostra como fazer uma consulta à API pelo nome da cidade



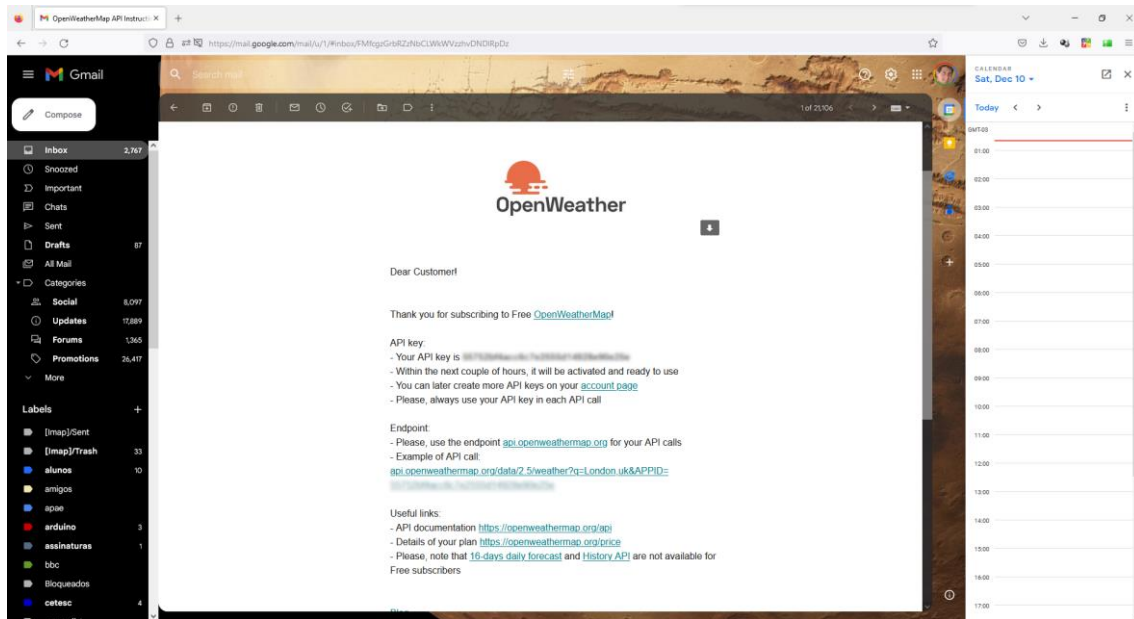
- d. Para usar a API é necessário obter uma chave e para recebê-la é preciso se cadastrar no *site*, neste [endereço](https://home.openweathermap.org/users/sign_up).

The screenshot shows the 'Create New Account' page on the OpenWeatherMap website. The browser address bar shows the URL https://home.openweathermap.org/users/sign_up. The form includes input fields for a username (filled with 'herculano.debiasi'), an email address (filled with 'herculano.debiasi@gmail.com'), and two password fields (both masked with '*****'). Below the form, there is a paragraph of text explaining the use of user information for management and communication purposes, with a link to the 'Privacy Policy'. There are two checked checkboxes: 'I am 16 years old and over' and 'I agree with Privacy Policy, Terms and conditions of sale and Websites terms and conditions of use'. Below these, there is a section for consent to receive communications from the OpenWeather Group, with three unchecked checkboxes: 'System news (API usage alert, system update, temporary system shutdown, etc)', 'Product news (change to price, new product features, etc)', and 'Corporate news (our life, the launch of a new service, etc)'. At the bottom of the form, there is a reCAPTCHA widget with the text 'Não sou um robô' and a 'Create Account' button.

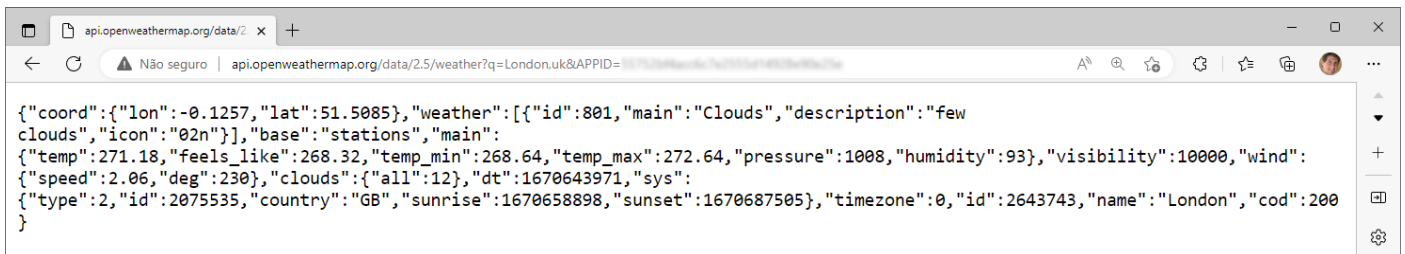
- e. É necessário confirmar o endereço respondendo o *e-mail* enviado.



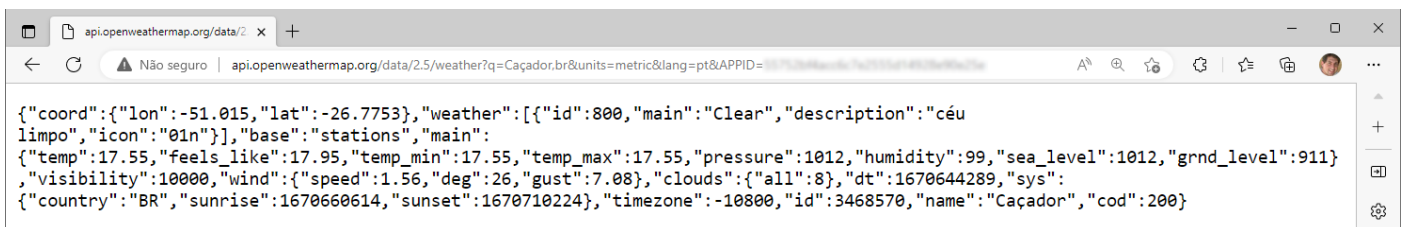
- f. Após isso um outro *e-mail* será enviado com a chave da API.



- g. No *e-mail* recebido, clique no *link* logo abaixo de '*example of API call*'. Demora alguns minutos até a chave ser ativada. Após isso deverá aparecer um retorno semelhante ao abaixo.



- h. Experimente alterar o nome da cidade e acrescente também os parâmetros `units=metric&lang=pt` a fim de adequar as unidades de medida e também as mensagens para português.



- i. Crie uma página HTML que contenha um formulário solicitando o nome da cidade e um botão; este, ao ser clicado, chamará uma função, que por sua vez fará uma requisição `GET` à essa API.
- j. Para fazer esta requisição `GET` em JS baseie-se em qualquer método visto nas aulas. Por exemplo, a apresentação '[aula 32b - XML \(02.12.2022\).pptx](#)' mostra várias formas diferentes (Ajax, jQuery e API Fetch), além da biblioteca Axios visto na aula de ontem. Você deverá montar uma URL semelhante à mostrada acima passando o nome da cidade entrada no formulário.
- k. Mostre o resultado da descrição e temperatura em uma `<div>`.