

Informações: O CPTEC-INPE possui serviços de previsão do tempo que podem ser consumidos usando apenas uma URL, a lista pode ser visualizada em <http://servicos.cptec.inpe.br/XML/>. Dentre as opções existe a previsão para 4 dias que está disponível pela URL <http://servicos.cptec.inpe.br/XML/cidade/4963/previsao.xml>.

O número 4963 é o código da cidade de São José dos Campos na base de dados do serviço do CPTEC. O código de cada cidade pode ser obtido usando a seguinte URL, onde o nome do município pode ser parcial:

<http://servicos.cptec.inpe.br/XML/listaCidades?city=sao%20jose>

Observação: o nome da cidade não pode conter caracteres acentuados e cedilha.

Objetivo: Fazer uma aplicação React para reproduzir o exemplo a seguir.

Nome

caç

Cidades

Caçapava - SP ▼

| Caçapava - SP (09/03/2020) | | | | |
|----------------------------|---------------------------|---------------------------------|---------------------------------|-------------------|
| Dia | 09/03/2020 | 10/03/2020 | 11/03/2020 | 12/03/2020 |
| Condições do tempo | Pancadas de Chuva a Tarde | Poss. de Panc. de Chuva a Tarde | Poss. de Panc. de Chuva a Tarde | Predomínio de Sol |
| Temp. máxima | 29 | 30 | 30 | 30 |
| Temp. mínima | 16 | 18 | 18 | 19 |
| Índice ultravioleta | 10.0 | 11.0 | 11.0 | 11.0 |

Considere os seguintes requisitos:

- O usuário deverá digitar parte do nome da cidade no campo Nome. O sistema deverá obter a lista de cidades no serviço do CPTEC e carregar no campo Cidades. Observação: o usuário precisará fornecer pelo menos 3 caracteres para o sistema fazer a busca;
- O usuário deverá selecionar uma cidade no campo Cidades. O sistema deverá obter a previsão de 4 dias para cidade no serviço do CPTEC e carregar numa tabela. Observação: a tabela só estará visível se o usuário já tiver escolhido uma cidade, isto é, a página não é carregada com uma tabela.

Para fazer a atividade considere as seguintes dicas:

- Criar o projeto:** Para criar o projeto React é necessário ter a ferramenta [create-react-app](#) instalada:

```
create-react-app -V
```

Caso não tenha ela instalada, então utilize o comando a seguir para instalar:

```
npm install -g create-react-app
```

Para criar um projeto de nome [atv-react](#): no CMD acesse a pasta que você desejar e digite o comando a seguir:

```
npx create-react-app atv-react
```

No CMD entre dentro da pasta [atv-react](#) e use o comando a seguir para subir a aplicação:

```
cd atv-react
```

```
npm start
```

- Requisição HTTP:** utilize a API Fetch do JavaScript para fazer requisições HTTP. Como exemplo teste o código a seguir no console do Google Chrome, e veja que a função global fetch retorna uma Promise.

```
let nome = 'cac';
fetch(`http://servicos.cptec.inpe.br/XML/listaCidades?city=${nome}`, { method: "GET" })
```

```
.then(response => console.log(response) )
.catch(erro => console.log(erro) );
```

Veja o resultado no console do Google Chrome:

```
> let nome = 'cac';
fetch(`http://servicos.cptec.inpe.br/XML/listaCidades?city=${nome}`, { method: "GET" })
.then(response => console.log(response) )
.catch(erro => console.log(erro) );
< ▶ Promise {<pending>}
▼ Response {type: "cors", url: "http://servicos.cptec.inpe.br/XML/listaCidades?city=cac", redirected: false, status: 200, ok: true, ...}
  type: "cors"
  url: "http://servicos.cptec.inpe.br/XML/listaCidades?city=cac"
  redirected: false
  status: 200
  ok: true
  statusText: "OK"
  ▶ headers: Headers {}
  body: (...)
  bodyUsed: false
  ▶ __proto__: Response
```

O método `text` pode ser utilizado para obter o corpo da resposta, mas precisaremos incluir outro `then` na Promise. Enquanto a página está codificada em UTF-8, o resultado possui codificação ISO-8859-1, então os caracteres especiais ficarão estranhos.

```
let nome = 'joao';
fetch(`http://servicos.cptec.inpe.br/XML/listaCidades?city=${nome}`, { method: "GET" })
.then(response => response.text() )
.then(txt => console.log(txt) )
.catch(erro => console.log(erro) );
```

```
<?xml version='1.0' encoding='ISO-8859-1'?><idades><cidade><nome>João Alfredo</nome><uf>PE</uf> VM2160:4
<id>2791</id></cidade><cidade><nome>João Câmara</nome><uf>RN</uf><id>2792</id></cidade><cidade><nome>João
Costa</nome><uf>PI</uf><id>2793</id></cidade><cidade><nome>João Dias</nome><uf>RN</uf><id>2794</id></cidade>
<cidade><nome>João Dourado</nome><uf>BA</uf><id>2795</id></cidade><cidade><nome>João Lisboa</nome>
<uf>MA</uf><id>2796</id></cidade><cidade><nome>João Monlevade</nome><uf>MG</uf><id>2797</id></cidade>
<cidade><nome>João Neiva</nome><uf>ES</uf><id>2798</id></cidade><cidade><nome>João Pessoa</nome><uf>PB</uf>
<id>231</id></cidade><cidade><nome>João Pinheiro</nome><uf>MG</uf><id>2799</id></cidade><cidade><nome>João
Ramalho</nome><uf>SP</uf><id>2800</id></cidade></idades>
```

A classe `TextDecoder` (<https://developer.mozilla.org/en-US/docs/Web/API/TextDecoder>) converte a codificação, mas ela pede como entrada uma `ArrayBuffer` (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/ArrayBuffer), por este motivo usou-se o método `arrayBuffer`.

```
let nome = 'joao';
fetch(`http://servicos.cptec.inpe.br/XML/listaCidades?city=${nome}`, { method: "GET" })
.then(response => response.arrayBuffer())
.then(buffer => {
  let decoder = new TextDecoder("iso-8859-1");
  return decoder.decode(buffer);
})
.then(txt => console.log(txt) )
.catch(erro => console.log(erro) );
```

Como o resultado da requisição é um texto no formato XML, então precisaremos converter de string para objeto XML.

```
let nome = 'joao';

fetch(`http://servicos.cptec.inpe.br/XML/listaCidades?city=${nome}`)
{ method: "GET" })
.then(response => response.arrayBuffer())
.then(buffer => {
  let decoder = new TextDecoder("iso-8859-1");
  return decoder.decode(buffer);
})
.then(str => {
  let parser = new window.DOMParser();
  let xml = parser.parseFromString(str, "text/xml");
  console.log(xml);
})
.catch(err => console.log(err));
```

[illegible]

Para percorrer o objeto XML precisaremos usar alguns dos seus métodos. O código a seguir cria um array com os nomes das cidades.

```

let nome = 'joao';

fetch(`http://servicos.cptec.inpe.br/XML/listaCidades?city=${nome}`, { method: "GET" })
  .then(response => response.arrayBuffer())
  .then(buffer => {
    let decoder = new TextDecoder("iso-8859-1");
    return decoder.decode(buffer);
  })
  .then(str => {
    let parser = new window.DOMParser()
    let xml = parser.parseFromString(str, "text/xml");
    //obter a tag cidades
    let cidades = xml.getElementsByTagName('cidades')[0];
    let lista = [];
    cidades.childNodes.forEach((obj, index) => {
      lista.push(obj.childNodes[0].childNodes[0].nodeValue)
    });
    console.log(lista);
  })
  .catch(erro => console.log(erro));

```

- iii. **Formatar a data:** nos dados retornados pelo CPTEC a data está no formato aaaa-mm-dd, então use o código a seguir para alterar a formatação.

```
formatData = data => {  
  let d = data.split('-');  
  return d[2] + '/' + d[1] + '/' + d[0];  
}
```

- iv. **Formatar a condição do tempo:** a previsão do tempo diária é dada por sigla, por exemplo, ps (Predomínio de Sol), então é necessário usar os dados da tabela (<http://servicos.cptec.inpe.br/XML/#condicoes-tempo>) para converter. O método a seguir recebe uma sigla e retorna a descrição.

```
getTempo = sigla => {  
  return {  
    'ec': 'Encoberto com Chuvas Isoladas',  
    'ci': 'Chuvas Isoladas',  
    'c': 'Chuva',  
    'in': 'Instável',  
    'pp': 'Poss. de Pancadas de Chuva',  
    'cm': 'Chuva pela Manhã',  
    'cn': 'Chuva a Noite',  
    'pt': 'Pancadas de Chuva a Tarde',  
    'pm': 'Pancadas de Chuva pela Manhã',  
    'np': 'Nublado e Pancadas de Chuva',  
    'pc': 'Pancadas de Chuva',  
    'pn': 'Parcialmente Nublado',  
    'cv': 'Chuveiro',  
    'ch': 'Chuvoso',  
    't': 'Tempestade',  
    'ps': 'Predomínio de Sol',  
    'e': 'Encoberto',  
    'n': 'Nublado',  
    'cl': 'Céu Claro',  
    'nv': 'Nevoeiro',  
    'g': 'Gedada',  
    'ne': 'Neve',  
    'nd': 'Não Definido',  
    'pnt': 'Pancadas de Chuva a Noite',  
    'psc': 'Possibilidade de Chuva',  
    'pcm': 'Possibilidade de Chuva pela Manhã',  
    'pct': 'Possibilidade de Chuva a Tarde',  
    'pcn': 'Possibilidade de Chuva a Noite',  
    'npt': 'Nublado com Pancadas a Tarde',  
    'nnp': 'Nublado com Pancadas a Noite',  
    'ncn': 'Nublado com Poss. de Chuva a Noite',  
    'nct': 'Nublado com Poss. de Chuva a Tarde',  
    'ncm': 'Nubl. c/ Poss. de Chuva pela Manhã',  
    'npm': 'Nublado com Pancadas pela Manhã',  
    'npp': 'Nublado com Possibilidade de Chuva',  
    'vn': 'Variação de Nebulosidade',  
    'ct': 'Chuva a Tarde',  
    'ppn': 'Poss. de Panc. de Chuva a Noite',  
    'ppt': 'Poss. de Panc. de Chuva a Tarde',  
    'ppm': 'Poss. de Panc. de Chuva pela Manhã'  
  }[sigla];  
}
```