Trabalhando com APIs

Neste projeto serão usadas as APIs

- API de CEP
  https://viacep.com.br/
- API de Bandeiras dos Países
  https://countryflagsapi.com/
- API de Banco de Imagens
  https://unsplash.com/developers
- API de Clima
  https://openweathermap.org/api

Desenvolva a atividade a seguir que trabalha com diversas API. As explicações estão ao decorrer do documento.

Parte I - Design

1) Incorpore o link do Boostrap

https://getbootstrap.com/docs/5.2/getting-started/introduction/

```
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css">
```

2) Incorpore o link do ícone do FontAweasome

```
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.2/css/all.min.css" integrity="sha512-
1sCRPdkRXhBV2PBLUdRb4tMg1w2YPf37qatUFeS7zlBy7jJI8Lf4VHwWfZZfpXtYSLy85pkm9GaYVYMfw5BC1A=
=" crossorigin="anonymous" referrerpolicy="no-referrer" />
```

3) Coloque um elemento Detail

```
<details> <summary> O que é um APP?  </summary>
    <p> A API faz a integração com os sistemas, fornecendo recursos (serviços).  </p>
 </details>
```

4) Crie as linhas usando as classes row e col do Boostrap, conforme exemplo abaixo. Dessa forma será definido automaticamente o espaço entre as colunas:

```
 <div class="row">
        <div class="col">
                 <label for="cidade">Cidade</label>
                <input type="texto" class="form-control" name="cidade" id="cidade" >
        </div>
        <div class="col">
                   ...
        </div>
        <div class="col">
          <span> Verificar Logradouro</span>
          <input type = "button" id="pesquisaCep" class="btn btn-outline-success" value="Pesquisar
CEP">
       </div>
    </div>
```

** As classes de Input do formulário são do Bootstrap ---- class="form-control"

** Nesta linha é usada a classe de botão do Boostrap

```
  <input type="button" id="verificarClima" value="Verificar Clima" class="btn btn-primary">
```

** Perceba que **não** foi linkada a função do JS direto ao botão... Dessa vez será feita a incorporação direto pelo JS, conforme item 6)

5) Crie uma div para exibir os resultados, como abaixo exemplo:

```
<label for="temperatura">Temperatura: <span id="temperatura"></span></label>
<label for="vento">Vento: <span id="vento"></span></label>
<img id="icone" src="" alt="Ícone do tempo">
<label id="cidadeTemp"></label>  <img id="imgPais" src="" alt="País">
<img id="imgCidade" src="" alt="Clima na cidade">
```

6) Acessando a API VIACEP e buscando os dados:

a) Para adicionar um evento ao botão:

Primeiro deve pegar o elemento botão, ou outro elemento, que queira vincular o evento click (ou outro) e adicionar o evento:
Conforme aplicado no código abaixo:

```
document.getElementById("pesquisaCep").addEventListener('click', PesquisarCep)

async function PesquisarCep()
    {
      let valor= document.getElementById('cep').value
      cep = valor.replace(/\D/g,'')  //remove todas as entradas que não forem dígitos
      if (cep != "")
      {
        const apiViaCep = 'https://viacep.com.br/ws/'+ cep + '/json/'

        const resCep = await fetch(apiViaCep)
        const data = await resCep.json()

        if (data.erro == true)
          alert("CEp Inválido")
        else
          document.getElementById("rua").value= `${data.logradouro} - ${data.bairro} -
${data.localidade} - ${data.uf} `
      }
      else
        alert("Formato de CEP inválido.");
    })
```

b) Para usar um microserviço (API externa) é necessário esperar o retorno da requisição e por isso é necessário usar a programação assíncrona.
No caso abaixo fará a requisição e ficará esperando a resposta.

```
const apiViaCep = 'https://viacep.com.br/ws/'+ cep + '/json/'
const resCep = await fetch(apiViaCep)
```

**Depois leia https://developer.mozilla.org/pt-BR/docs/Web/API/Fetch_API/Using_Fetch
https://developer.mozilla.org/en-US/docs/Web/API/Response/json

c) Conversão dos dados JSON e dados a serem processados no JS

```
const data = await resCep.json()
```

O nome da varíavel que receberá o retorno dos dados pode ser qualquer um, normalmente definimos **data**

d) Tratar os erros e retornos

O retorno do VIACEP pode ser de **erro {erro: true}**. Para testar se não achou o CEP pode ser usada uma estrutura de decisão IF

```
if (data.erro == true)
     alert("CEp Inválido")
```

7) Acessando a API OpenWeatherMap de Clima
Para fazer uso da API OpenWeatherMap é necessário fazer um cadastro.

- https://openweathermap.org/price -- gratuita até 100 requisições por dia

e pegar o seu número de chave de registro da API. Ele deverá ser inserido na variável apiKey para uso da API.

```
apiKey = "----------";
** Depois códigos estão no Item 10)
```

8) Acessando a API Unsplah de Imagem
No projeto atual apenas foi usado o endereço do BD de Imagem com o nome da cidade, mas a API tem vários recursos e podem ser acessados por sua chave criada.
https://unsplash.com/developers

No caso do projeto atual foi criada uma variável

```
apiBDImagemUnsplash = "https://source.unsplash.com/1600x900/?";
```

E… depois alterado no elemento de imagem HTML o atributo "src" de forma que pegue o nome da cidade digitada e a concatene com o endereço do Unsplash

```
document.getElementById('imgCidade').setAttribute('src',
apiBDImagemUnsplash+cidade);
```

9) Acessando a API CountryFlag com ícones das bandeiras dos países

Aqui foi feito igual ao país, passado a abreviação do país ao endereço da API e alterado o "src" da imagem da Bandeira do site.

```
apiBandeiraPais = "https://countryflagsapi.com/png/"
document.getElementById("imgPais").setAttribute("src",
apiBandeiraPais+data.sys.country);
```

10)Acesssando a API do Tempo

```
** Inicialmente foram criadas as variável para facilitar o Desenvolvimento
apiKey = "----------"
apiBDImagemUnsplash = "https://source.unsplash.com/1600x900/?"
apiBandeiraPais = "https://countryflagsapi.com/png/"
cidade = document.querySelector("input[name='cidade']").value

**Depois foram pegos os elementos do HTML
tempElement = document.querySelector("#temperatura");
climaIconElement = document.querySelector("#clima-icon");
dadosTemp = document.querySelector(".dadosTemperatura")
dadosLog = document.querySelector(".dadosLogradouro")

**Lembrem-se que para pegar os elementos HTML pode ser usados vários métodos JS,
como querySelector, getElementById.
**Lembrem ainda que para pegar a classe é .nome da classe, e pelo identificador #id

**Aqui tem o código do botão que verificar o tempo da cidade digitada:

document.querySelector("#verificarClima").addEventListener("click", async (e) => {
  e.preventDefault()
```

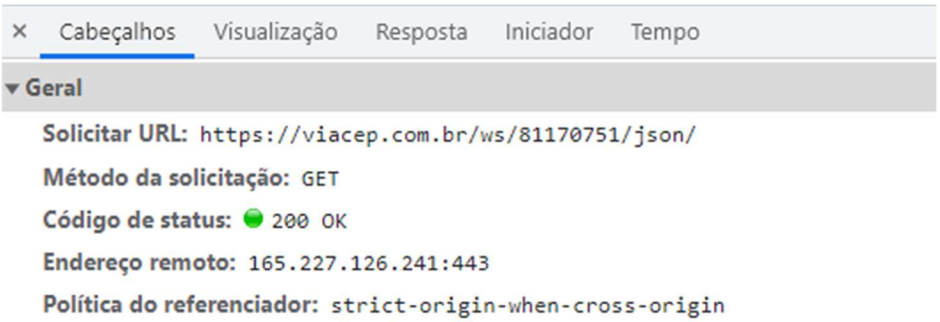
```
  const apiClimaURL =
`https://api.openweathermap.org/data/2.5/weather?q=${cidade}&units=metric&appid=${ap
iKey}&lang=pt_br`

  const res = await fetch(apiClimaURL)
  const data = await res.json()
  if (data.cod === "404") {
    alert("Cidade Inválida")
    return
  }

  temperatura.innerText = parseFloat(data.main.temp)+ "º"; //poderia ser &deg;C
  humidade.innerText = parseFloat(data.main.humidity)+ "%"
  vento.innerText = parseFloat(data.wind.speed)+ "km/h"
  tempo.innerText = data.weather[0].description;
  document.getElementById("cidadeTemp").innerText = cidade;
  document.getElementById("icone").setAttribute("src",`http://openweathermap.org/img
/wn/${data.weather[0].icon}.png`);
  document.getElementById("imgPais").setAttribute("src",
apiBandeiraPais+data.sys.country);
  document.getElementById('imgCidade').setAttribute('src',
apiBDImagemUnsplash+cidade);
});
```

11)Observe os retornos das API

** Perceba que o retorno da API do tempo retornou diversos dados em formato de objeto

**Observe que para pegar o ícone e a descrição foi colocado o índice do array
weather [{ }] --- veja os colchetes que definem um array

** Perceba que o retorno da API do ViaCEP retornou os dados no formato JSON porque foi colocado no endereço o formato de retorno. Leia a API e veja que tem outras formas de trabalhar com o retorno.



{cep: "81170-751", logradouro: "Rua Sady Silva", complemento: "", bairro: "Cidade Industrial",…}
bairro: "Cidade Industrial"
cep: "81170-751"
complemento: ""
ddd: "41"
gia: ""
ibge: "4106902"
localidade: "Curitiba"
logradouro: "Rua Sady Silva"
siafi: "7535"
uf: "PR"

**Para pegar os dados do tempo e do CEP foram:

```
data.main.temp  ou data.weather[0].description
data.logradouro
```

Veja que os Objetos JSon de retorno da API são diferentes. O da API do tempo é formato por outros objetos e também um array de objetos. Tem que apontar corretamente, indicar corretamente o dado desejado.

{cep: "81170-751", logradouro: "Rua Sady Silva", complemento: "", bairro: "Cidade Industrial",…}

```
▶ main: {temp: 12.03, feels_like: 11.56, temp_min: 11.16, temp_max: 12.25, pressure: 1021, humidity: 87}
  name: "Curitiba"
▶ sys: {type: 2, id: 67576, country: "BR", sunrise: 1667723128, sunset: 1667770567}
  timezone: -10800
  visibility: 10000
▼ weather: [{id: 803, main: "Clouds", description: "nublado", icon: "04n"}]
  ▶ 0: {id: 803, main: "Clouds", description: "nublado", icon: "04n"}
```