

# Apostila: Consumo de API com Múltiplo Select de Estado e Cidade e Busca de CEP

## Introdução

O consumo de APIs (Application Programming Interfaces) é uma habilidade essencial para desenvolvedores front-end e back-end. Nesta apostila, vamos aprender como integrar uma API pública para criar um sistema de seleção dinâmica de estados e cidades, além de implementar uma busca de CEP. Utilizaremos HTML, CSS e JavaScript para construir a interface e realizar as requisições à API.

## Objetivo

O objetivo desta apostila é capacitar os alunos a:

- Compreender o funcionamento de APIs RESTful.
- Realizar requisições HTTP utilizando JavaScript (fetch API).
- Popular campos de seleção (select) dinamicamente.
- Implementar uma busca de CEP e exibir os dados retornados pela API.

## Pré-requisitos

- Conhecimento básico de HTML, CSS e JavaScript.
  - Noções sobre requisições HTTP (GET, POST) e manipulação de JSON.
- 

# Capítulo 1: Configurando o Ambiente

## 1.1 Estrutura do Projeto

Crie uma pasta para o projeto e organize os arquivos da seguinte forma:

```
projeto-multiplo-select/  
├── index.html  
├── style.css  
└── script.js
```

## 1.2 HTML Básico

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
  <meta charset="UTF-8">  
  <title>Consumo de API - Estados, Cidades e CEP</title>  
  <link rel="stylesheet" href="style.css">  
</head>
```

```
<body>
  <h1>Seleção de Estado e Cidade</h1>
  <label for="estado">Estado:</label>
  <select id="estado" onchange="carregarCidades()">
    <option value="">Selecione o estado</option>
  </select>

  <label for="cidade">Cidade:</label>
  <select id="cidade">
    <option value="">Selecione a cidade</option>
  </select>

  <h2>Busca de CEP</h2>
  <input type="text" id="cep" placeholder="Digite o CEP" maxlength="8">
  <button onclick="buscarCEP()">Buscar</button>

  <div id="resultadoCEP"></div>

  <script src="script.js"></script>
</body>
</html>
```

---

# Capítulo 2: Estilizando com CSS

## 2.1 Estilos Básicos

```
body {
  font-family: Arial, sans-serif;
  padding: 20px;
  background-color: #f2f2f2;
}
select, input {
  padding: 10px;
  margin: 10px 0;
  width: 200px;
}
button {
  padding: 10px 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  cursor: pointer;
}
button:hover {
  background-color: #45a049;
}
#resultadoCEP {
  margin-top: 20px;
  padding: 10px;
  border: 1px solid #ddd;
  background-color: white;
}
```

---

# Capítulo 3: Consumindo a API com JavaScript

## 3.1 Carregando Estados

```
window.onload = () => carregarEstados();

function carregarEstados() {
  fetch('https://servicodados.ibge.gov.br/api/v1/localidades/estados')
    .then(response => response.json())
    .then(estados => {
      const selectEstado = document.getElementById('estado');
      estados.sort((a, b) => a.nome.localeCompare(b.nome));
      estados.forEach(estado => {
        const option = document.createElement('option');
        option.value = estado.id;
        option.text = estado.nome;
        selectEstado.add(option);
      });
    })
    .catch(error => console.error('Erro ao carregar estados:', error));
}
```

## 3.2 Carregando Cidades

```
function carregarCidades() {
  const estadold = document.getElementById('estado').value;
  if (!estadold) return;

  fetch(`https://servicodados.ibge.gov.br/api/v1/localidades/estados/${estadold}/municipios`)
    .then(response => response.json())
    .then(cidades => {
      const selectCidade = document.getElementById('cidade');
      selectCidade.innerHTML = '<option value="">Selecione a cidade</option>';
      cidades.forEach(cidade => {
        const option = document.createElement('option');
        option.value = cidade.id;
        option.text = cidade.nome;
        selectCidade.add(option);
      });
    })
    .catch(error => console.error('Erro ao carregar cidades:', error));
}
```

## 3.3 Busca de CEP

```
function buscarCEP() {
  const cep = document.getElementById('cep').value;
  if (!cep) return alert('Digite um CEP válido!');

  fetch(`https://viacep.com.br/ws/${cep}/json/`)
    .then(response => response.json())
    .then(dados => {
      const resultado = document.getElementById('resultadoCEP');
    });
}
```

```
if (dados.erro) {
  resultado.textContent = 'CEP não encontrado!';
} else {
  resultado.innerHTML = `
    <p><strong>Endereço:</strong> ${dados.logradouro}</p>
    <p><strong>Bairro:</strong> ${dados.bairro}</p>
    <p><strong>Cidade:</strong> ${dados.localidade}</p>
    <p><strong>Estado:</strong> ${dados.uf}</p>
  `;
}
})
.catch(error => console.error('Erro ao buscar CEP:', error));
}
```

---

## Conclusão

Neste material, exploramos como consumir APIs públicas para popular selects de estados e cidades, além de realizar uma busca de CEP. Pratique e adapte o código para suas necessidades!