Aluno(a): Alice Barros Viana

LISTA DE EXERCÍCIOS I

1. A Entrega Padrão:

Crie uma requisição do tipo GET.

Tente buscar o CEP da nossa faculdade (ou um famoso, como o da Sé em SP: 01001000).

Pergunta: Qual foi o Status Code retornado? O JSON veio completo?

O status Code foi de 200 OK e o json veio completo com todas as informações.

2. O Caso do CEP Inexistente:

Tente fazer um GET para um CEP que não existe (ex: 99999999).

Desafio: Observe o JSON retornado.

O ViaCEP retorna um erro 404 ou ele retorna um JSON com um campo "erro": true? Por que você acha que eles escolheram essa estratégia?

Ele retornou um campo com "erro": true, porque normalmente o 404 é utilizado para indicar que a URL ou recurso não existe. No caso, a API está funcionando corretamente; apenas o CEP informado é inválido ou não está cadastrado. Por isso, ela retorna esse erro para sinalizar que a API respondeu, mas o código informado não corresponde a um CEP válido.

3. Mudando o Formato (O Camaleão):

O ViaCEP é incrível porque ele aceita outros formatos além do JSON. Tente mudar o final da URL de /json/ para /xml/.

Pergunta: O que mudou na visualização dos dados? Qual formato parece mais fácil de ler no C#?

O formato XML é mais verboso do que o JSON, que é mais fácil de ler. Acredito que o formato JSON seja mais simples de interpretar no C#, justamente por sua estrutura mais enxuta.

PASSO 3: O RELATÓRIO DO ENGENHEIRO

Após realizar os testes, responda:

1. Verbo HTTP: Qual método você usou? Por que não usamos POST para consultar um CEP?

Usei o método GET porque ele serve para buscar dados em uma API. Já o método POST é geralmente utilizado para enviar informações ao servidor, criando ou modificando recursos. Como consultar um CEP não envolve alteração de dados, o GET é o mais adequado.

2. Análise de Dados: Copie o JSON do seu CEP e identifique:


```
{
  "cep": "01001-000",
  "logradouro": "Praça da Sé",
  "complemento": "lado ímpar",
  "unidade": "",
  "bairro": "Sé",
  "localidade": "São Paulo",
  "uf": "SP",
  "estado": "São Paulo",
  "regiao": "Sudeste",
  "ibge": "3550308",
  "gia": "1004",
  "ddd": "11",
  "siafi": "7107"
}
```


Qual é a Chave (Key) que guarda o nome da cidade?

A chave que guarda o nome da cidade é chamada de localidade.

Qual é a Chave que guarda o nome da rua?

A chave que guarda o nome da rua é chamada de logradouro.

3. **Integração C#:** Se você fosse criar uma classe em C# para recebe esses dados, como ficaria a sua propriedade para a chave localidade? Use o que aprendemos sobre

[JsonPropertyName].

```
using System.Text.Json.Serialization;

public class DistribuídaLog
{
   [JsonPropertyName("localidade")]
   public string Localidade { get; set; }
}
```