Passo 2

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

Resposta: O status foi **200 OK**. O json veio completo.

2 - 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?

Resposta: Ele retorna "erro": "true". Talvez para padronizar o retorno e não travar a aplicação com um erro, apenas notificar o usuário.

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

Resposta: O arquivo em xml passou a ser organizado em tags. Pelo layout simplificado acredito que o C# atue melhor com o Json.

Passo 3

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

consultar um CEP?

Resposta: Utilizei o método GET, pois estava pegando uma informação. O método POST é utilizado para lançar uma informação.

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

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

o Qual é a Chave que guarda o nome da rua? A chave é Logradouro

{

"cep": "30640-070",

"logradouro": "Avenida Afonso Vaz de Melo",

"complemento": "até 1999/2000",

"unidade": "",

"bairro": "Barreiro",

"localidade": "Belo Horizonte",

"uf": "MG",

"estado": "Minas Gerais",

"regiao": "Sudeste",

"ibge": "3106200",

"gia": "",

"ddd": "31",

"siafi": "4123"

}

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

Resposta:

```
public class Endereco

{    [JsonPropertyName("localidade")]

   public string Localidade { get; set; }

}
```