

Centro Universitário UNA  
Sistemas Distribuídos e Mobile  
Professor: Daniel Henrique Matos de Paiva  
Aluno: Rafael Felipe Nunes Dias  
Lista de Exercícios I  
Atividade: Operação Rastreio e Logística  
Passo 2 – Mão na Massa  
1) A Entrega Padrão

Foi realizada uma requisição do tipo GET para o seguinte endpoint:

<https://viacep.com.br/ws/01001000/json/>  
Status Code retornado: 200 OK  
O JSON retornou completo com os seguintes dados:  
Json

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

2) O Caso do CEP Inexistente  
Requisição realizada:

<https://viacep.com.br/ws/99999999/json/>  
Status Code retornado: 200 OK

JSON retornado:

```
Json  
{  
    "erro": true  
}
```

O serviço não retorna erro 404. Ele retorna 200 OK com "erro": true, pois a requisição foi processada corretamente pelo servidor, mas o CEP informado não existe na base de dados.

3) Mudando o Formato (JSON para XML)  
Requisição realizada:

<https://viacep.com.br/ws/01001000/xml/>  
O retorno passou a ser em formato XML:  
Xml  
<xmlcep>  
<cep>01001-000</cep>  
<logradouro>Praça da Sé</logradouro>  
<complemento>lado ímpar</complemento>  
<bairro>Sé</bairro>

```
<localidade>São Paulo</localidade>
<uf>SP</uf>
<ibge>3550308</ibge>
<gia>1004</gia>
<ddd>11</ddd>
<siafi>7107</siafi>
</xmlcep>
```

O que mudou foi apenas o formato de representação dos dados.

O formato JSON é mais simples e mais fácil de trabalhar no C#, principalmente utilizando System.Text.Json.

Passo 3 – Relatório do Engenheiro

1) Verbo HTTP

O método utilizado foi GET.

Não utilizamos POST porque estamos apenas consultando dados.

O método GET é utilizado para buscar informações sem modificar o servidor.

2) Análise do JSON

No JSON retornado:

A chave que guarda o nome da cidade é: localidade

A chave que guarda o nome da rua é: logradouro

3) Integração C#

Classe C# que representa o JSON:

Csharp

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

```
public class Endereco
{
    [JsonPropertyName("cep")]
    public string Cep { get; set; }

    [JsonPropertyName("logradouro")]
    public string Logradouro { get; set; }

    [JsonPropertyName("complemento")]
    public string Complemento { get; set; }

    [JsonPropertyName("bairro")]
    public string Bairro { get; set; }

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

    [JsonPropertyName("uf")]
    public string Uf { get; set; }

    [JsonPropertyName("ibge")]
    public string Ibge { get; set; }
}
```

```
[JsonPropertyName("gia")]
public string Gia { get; set; }

[JsonPropertyName("ddd")]
public string Ddd { get; set; }

[JsonPropertyName("siafi")]
public string Siafi { get; set; }
}
```