

Centro Universitário UNA

Sistemas Distribuídos e Mobile

Professor Daniel Henrique Matos de Paiva

Lista – 01

Nome: Pedro Lucas Dias Amorim de Deus

RA: 324122649

## Questão 1



The screenshot shows a REST client interface with the 'Response' tab selected. The status is '200 OK' with a response time of '481ms' and a size of '290B'. The JSON response is displayed as follows:

```
1  {
2    "cep": "01001-000",
3    "logradouro": "Praça da Sé",
4    "complemento": "lado ímpar",
5    "unidade": "",
6    "bairro": "Sé",
7    "localidade": "São Paulo",
8    "uf": "SP",
9    "estado": "São Paulo",
10   "regiao": "Sudeste",
11   "ibge": "3550308",
12   "gia": "1004",
13   "ddd": "11",
14   "siafi": "7107"
15 }
```

O servidor retornou status **200 OK** e o JSON veio completo com todas as informações do endereço.

## Questão 2



The screenshot shows a REST client interface with the 'Response' tab selected. The status is '200 OK' with a response time of '460ms' and a size of '20B'. The JSON response is displayed as follows:

```
1  {
2    "erro": "true"
3  }
```

Ele retorna status 200 OK, porém envia no JSON o campo "erro": true.

### Questão 3

JSON -> Formato de um dicionário.

XML -> Formato em tags.

O JSON é mais fácil de manipular no C# porque pode ser convertido diretamente em objetos usando System.Text.Json, enquanto o XML exige configurações extras e classes específicas.

Questão 1 – Método usado foi o GET, o método POST não foi utilizado, pois usamos ele para cadastrar ou enviar informações, nesse caso o CEP foi apenas consultado, ele apenas retorna dados.

Questão 2 – A chave para cidade é a “localidade” e para a rua é “logradouro”.

Questão 3 -

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

```
public class Endereco
```

```
{
```

```
    [JsonPropertyName("localidade")]
```

```
    public string Localidade { get; set; }
```

```
    [JsonPropertyName("logradouro")]
```

```
    public string Logradouro { get; set; }
```

```
}
```