

# ATIVIDADE CONSULTA VIA CEP REQUISIÇÕES

## FELIPE JORGE RODRIGUES DE SOUSA

### QUESTÃO 01: CONVERTER JSON EM XML E IMPRIMIR A REQUISIÇÃO

```
01_Consumo_de_API_XML.py X
01_Consumo_de_API_XML.py > ...
1  import requests
2  import xml.etree.ElementTree as ET
3
4  url = 'https://viacep.com.br/ws/'
5  cep = 30140071
6  formato = '/xml/'
7
8  r = requests.get(url + str(cep) + formato)
9
10 if (r.status_code == 200):
11     # Converte a resposta JSON para XML como o método nativo
12     root = ET.fromstring(r.content)
13     xml_string = ET.tostring(root, encoding='utf-8', method='xml')
14
15     print()
16     print('XML : ', xml_string.decode())
17 else:
18     print("Não houve sucesso na requisição. Erro:" + r.status_code)
```

### TERMINAL:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
$ cls ; python -u "d:\felipe\Documents\UNA\SISTEMAS_DISTRIBUIDOS\Atividades\01\01_Consumo_de_API_XML.py"
bash: cls: command not found

XML : <xmlcep>
<cep>30140-071</cep>
<logradouro>Rua dos Aimorés</logradouro>
<complemento>de 971/972 a 1399/1400</complemento>
<bairro>Boa Viagem</bairro>
<localidade>Belo Horizonte</localidade>
<uf>MG</uf>
<ibge>3106200</ibge>
<gia />
<ddd>31</ddd>
<siafi>4123</siafi>
</xmlcep>

felipe@DESKTOP-SD0PEET MINGW64 /d/felipe/Documents/UNA/SISTEMAS_DISTRIBUIDOS/Atividades/01
$
```

## QUESTÃO 02: CONSULTAR 05 CEPS ITERANDO EM UM FOR IN.

```
02_Consumo_de_API_XML.py X
02_Consumo_de_API_XML.py > ...
1  import requests
2  import xml.etree.ElementTree as ET
3
4  url = 'https://viacep.com.br/ws/'
5  cep = 30140071
6  formato = '/xml/'
7
8  for cont in range(5):
9      r = requests.get(url + str(cep + cont) + formato)
10
11     if (r.status_code == 200):
12         # Converte a resposta JSON para XML como o método nativo
13         root = ET.fromstring(r.content)
14         xml_string = ET.tostring(root, encoding='utf-8', method='xml')
15
16     print()
17     print('XML : ', xml_string.decode())
18
```

## TERMINAL

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

$ cls ; python -u "d:\felipe\Documents\UNA\SISTEMAS_DISTRIBUIDOS\
Atividades\01\01_Consumo_de_API_XML.py"
bash: cls: command not found

XML : <xmlcep>
<cep>30140-071</cep>
<logradouro>Rua dos Aimorés</logradouro>
<complemento>de 971/972 a 1399/1400</complemento>
<bairro>Boa Viagem</bairro>
<localidade>Belo Horizonte</localidade>
<uf>MG</uf>
<ibge>3106200</ibge>
<gia />
<ddd>31</ddd>
<siafi>4123</siafi>
</xmlcep>

XML : <xmlcep>
<cep>30140-072</cep>
<logradouro>Rua dos Aimorés</logradouro>
<complemento>de 1401/1402 a 1799/1800</complemento>
<bairro>Lourdes</bairro>
<localidade>Belo Horizonte</localidade>
<uf>MG</uf>
<ibge>3106200</ibge>
<gia />
<ddd>31</ddd>
<siafi>4123</siafi>
</xmlcep>
```

```
XML : <xmlcep>
  <cep>30140-073</cep>
  <logradouro>Rua dos Aimorés</logradouro>
  <complemento>de 2761/2762 a 3299/3300</complemento>
  <bairro>Barro Preto</bairro>
  <localidade>Belo Horizonte</localidade>
  <uf>MG</uf>
  <ibge>3106200</ibge>
  <gia />
  <ddd>31</ddd>
  <siafi>4123</siafi>
</xmlcep>
```

```
XML : <xmlcep>
  <cep>30140-074</cep>
  <logradouro>Rua dos Aimorés</logradouro>
  <complemento>de 1801/1802 a 2369/2370</complemento>
  <bairro>Lourdes</bairro>
  <localidade>Belo Horizonte</localidade>
  <uf>MG</uf>
  <ibge>3106200</ibge>
  <gia />
  <ddd>31</ddd>
  <siafi>4123</siafi>
</xmlcep>
```

```
XML : <xmlcep>
  <cep>30140-075</cep>
  <logradouro>Rua dos Aimorés</logradouro>
  <complemento>de 531/532 a 969/970</complemento>
  <bairro>Funcionários</bairro>
  <localidade>Belo Horizonte</localidade>
  <uf>MG</uf>
  <ibge>3106200</ibge>
  <gia />
  <ddd>31</ddd>
  <siafi>4123</siafi>
</xmlcep>
```

5:31:05 PREFIXED\_CDDPREFET\_MINGUELA\_11/5-2019 (Downtown) / UNAC/CTC/TEMAS\_DISTRIBUIDOS (Atividade)

### QUESTÃO 03: CONSULTAR POR NOME DA RUA CÓDIGO

```
03_consulta_cidade_xml.py X
03_consulta_cidade_xml.py > ...
1  ~ import requests
2  import xml.etree.ElementTree as ET
3
4  ~ def consulta_endereco(estado, cidade, rua):
5      url = 'https://viacep.com.br/ws/'
6      formato = '/xml/'
7
8      r = requests.get(url + estado + '/' + cidade + '/' + rua + formato)
9
10 ~     if (r.status_code == 200):
11         # Converte a resposta XML para uma string
12         root = ET.fromstring(r.content)
13         xml_string = ET.tostring(root, encoding='utf-8', method='xml')
14         print()
15         print('XML : ', xml_string.decode())
16
17     estado = 'MG'
18     cidade = 'Belo Horizonte'
19     rua = 'Rua Rio de Janeiro'
20
21     consulta_endereco(estado, cidade, rua)
22
```

TERMINAL:( PRINT REDUZIDO, POIS SÃO VÁRIAS PÁGINAS DE RESPOSTAS DA REQUISIÇÃO)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

<\UNA\SISTEMAS_DISTRIBUIDOS\Atividades\01\03_consulta_cidade_xml.py"

XML :  <xmlcep>
<enderecos>
  <endereco>
    <cep>30160-040</cep>
    <logradouro>Rua Rio de Janeiro</logradouro>
    <complemento>até 393/394</complemento>
    <bairro>Centro</bairro>
    <localidade>Belo Horizonte</localidade>
    <uf>MG</uf>
    <ibge>3106200</ibge>
    <gia />
    <ddd>31</ddd>
    <siafi>4123</siafi>
  </endereco>
  <endereco>
    <cep>30160-913</cep>
    <logradouro>Rua Rio de Janeiro</logradouro>
    <complemento>909</complemento>
    <bairro>Centro</bairro>
    <localidade>Belo Horizonte</localidade>
    <uf>MG</uf>
    <ibge>3106200</ibge>
    <gia />
    <ddd>31</ddd>
    <siafi>4123</siafi>
  </endereco>
  <endereco>
    <cep>30160-915</cep>
    <logradouro>Rua Rio de Janeiro</logradouro>
    <complemento>985</complemento>
    <bairro>Centro</bairro>
    <localidade>Belo Horizonte</localidade>
    <uf>MG</uf>
    <ibge>3106200</ibge>
    <gia />
    <ddd>31</ddd>
```

## QUESTÃO 04: IMPRIMIR ERRO DA REQUISIÇÃO

```
04_print_erro_de_reqest_XML.py X
04_print_erro_de_reqest_XML.py > ...
1  import requests
2  import xml.etree.ElementTree as ET
3
4  url = 'https://viacep.com.br/abc/'
5  cep = 30140071
6  formato = '/xml/'
7
8  for cont in range(5):
9      r = requests.get(url + str(cep + cont) + formato)
10
11     if (r.status_code == 200):
12         # Converte a resposta JSON para XML como o método nativo
13         root = ET.fromstring(r.content)
14         xml_string = ET.tostring(root, encoding='utf-8', method='xml')
15
16         print()
17         print('XML : ', xml_string.decode())
18     else:
19         print("Não houve sucesso na requisição. Erro:" , r.status_code)
```

### TERMINAL:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
$ cls ; python -u "d:\felipe\Documents\UNA\SISTEMAS_DISTRIBUIDOS\Atividades\01\04_print_erro_de_reqest_XML.py"
bash: cls: command not found
Não houve sucesso na requisição. Erro: 404
Não houve sucesso na requisição. Erro: 404
Não houve sucesso na requisição. Erro: 404
Não houve sucesso na requisição. Erro: 404
Não houve sucesso na requisição. Erro: 404
felip@DESKTOP-SD0PEET MINGW64 /d/felipe/Documents/UNA/SISTEMAS_DISTRIBUIDOS/Atividades/01
$
```

## QUESTÃO 05: SALVAR O RESULTA DA REQUISIÇÃO EM UM ARQUIVO]

```
05_SALVANDO_CONSULTA.PY X
05_SALVANDO_CONSULTA.PY > ...
1  import requests
2  r = requests.get('http://www.google.com/search', params={'q': 'elson de abreu'})
3
4  if (r.status_code == 200):
5      print()
6      # print('Retorno : ', r.text)
7      #salvando a consulta em um arquivo
8      with open('result_consulta.txt','a') as arquivo :
9          arquivo.write(r.text)
10     print("A consulta foi gravada no arquivo com sucesso")
11 else:
12     print('Nao houve sucesso na requisicao.')
```

## ARQUIVO

```
result_consulta.txt
```

```
<!doctype html><html lang="pt-BR"><head><meta charset="UTF-8"><meta content="AmFmPme9tD7tVHZ5DTQ6/aRAYyKlIQDI/  
+Hwz5Y7OagTtDB3LA0tRMKtpQ8jCqkCmZ7HDuCpEwx7WwaIFBywoAAAByeYJvcmlnaW40IjoJdHRwciovL2dvb2dsZS5jZ2UuYnI6NDQzIiwvZWVhdHVzYS2I6Ikvx  
bmdBmItVXRpb25GcmFTZVRpbWluZyIsImVhcGlgeSYsMTGtOTVMzES05OWiaXNTdWJkb21haW40OnRydWV9" http-equiv="origin-trial"><meta  
content="/images/branding/google/1x/google_standard_color_128dp.png" itemprop="image"><title>elton de abreu - Pesquisa  
Google</title><script nonce="7to7SEr5g3AgYPnhgvto">(function(){  
2 document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");  
a="1"===c||"a"===c&&a.elements.q.value!=""?1:!1}else a=!1;a&&(b.preventDefault(),b.stopPropagation()),10);document.  
documentElement.addEventListener("click",function(b){var a;{for(a=b.target;a&&!a==document.documentElement;a=a.  
parentElement)if("A"===a.tagName)a="1"===a.getAttribute("data-noreferrer");break}a=!1;a&&b.preventDefault(),10;)).call(this);  
(function(){var a=window.performance;window.start=Date.now();{a;var b=window;if(a){var c=a.timing;if(c){var d=c.  
navigationStart,f=c.responseStart;if(f>d&&f<=window.start){window.start=f;b.wsrt=f-d;break}a}}a.now&&(b.wsrt=Math.floor(a.now  
()))};window.google=window.google||{};var h=function(g){g&&g.target.setAttribute("data-iml",String(Date.now()));};document.  
documentElement.addEventListener("load",h,10);google.rglh=function(){document.documentElement.removeEventListener("load",h,  
!0)};}).call(this);(function(){window.google.erd={jsr:1,bv:1970,de:true});})();(function(){var sdo=false;var mei=10;  
3 var h=this||self;var k,l=null==(k=h.mei)?k:1,n,p=null==(n=h.sdo)?n:10,q=0,r,t=google.erd,vt.jsr=google.ml=function(a,b,d,m,e)  
{e=void 0===e?2:e;b&&(r=a&&a.message);void 0===d&&(d={});d.cad="ple"+"google.ple"+aple+"google.aple;if(google.d)return  
google.dl(a,e,d),null;b=d;if(0>v){window.console&&console.error(a,b);if(-2===v)throw a;b=!1}else b=!1|a.message||"Error  
loading script"===a.message||q>l&&m!1?1:0;if(!b)return null;q+=d+d||{};b=encodeURIComponent;var c="/gen_204?atyp=i&ei="+b  
(google.kEI)+google.kEXPI&c+="&jexpid="+b(google.kEXPI));c+="&srcpg="+b(google.sn)+"&jsr="+b(vt.jsr)+"&bver="+  
4 b(t.bv);var f=a.lineNumber;void 0!==f&&(c+="&line="+f);var g=a.fileName;g&&(c+"<g_indexOf("-extension:")"&&(e=3),c+="&script="+b  
(g,f)&&g===window.location.href&&(f=document.documentElement.outerHTML.split("\n")[f],c+="&cad="+b(f?f.substring(0,300)":No  
script found."));google.ple&&1===google.ple&&(e=2);c+="&jsel="+e;for(var u in d)c+="&b(u,c)+";c+=b(d[u]);c+="&t"&&"emsg="  
+b(a.name+": "+a.message);c+c+"&jsst="+b(a.stack||"/N/A");12288<c.length&&(c<=c.substr(0,12288));a;c:m||google.log(0,""),a;  
return a;}window.onerror=function(a,b,d,m,e){r!=a&&(a=e instanceof Error?e>Error(a).void 0===d||"lineNumber"in a||a.  
lineNumber-d).void 0===b||"fileName"in a||a(fileName=b),google.ml(a,1,void 0,1,"SyntaxError"===a.name||"SyntaxError"===a.  
message.substring(0,11))|-1===a.message.indexOf("Script error")?3:0);r=null;p&&q>l&&(window.onerror=null)};})();(function()  
{var c=[];e=0;window.ping=function(c){!1===b.indexOf("&xz")&&(b+="&xz"+Date.now());var a=new Image,d=++c[c[d]-a.a.onerror=a.  
onload,a.onabort=function(){}delete c[d];a.src=b=u);}.call(this);</script><style>a{color:#1A0AD3;text-decoration:none;  
tap-highlight-color:rgba(0,0,0,.10)}a:visited{color:#4b1a18}a:hover{text-decoration:underline}img{border:0}html  
{font-family:arial,sans-serif;font-size:14px;line-height:20px;text-size-adjust:100%;color:#3C4043;word-wrap:break-word;  
background-color:#FFF;BRSMnc;background-color:#FFF;border-top:1px solid #dadce0;height:39px;overflow:hidden}.NRWV  
{height:51px;overflow-scrolling:touch;overflow-x:auto;overflow-y:hidden}.Uv67qb{box-pack:justify;font-size:12px;  
line-height:37px;justify-content:space-between;justify-content:space-between}.Uv67qb a.Uv67qb span{color:#70757a;  
display:block;flex:none;padding:0 16px;text-align:center;text-transform:uppercase;span.OXXUp{border-bottom:2px solid #4285f4;
```

## NAVEGADOR

