



DESENVOLVIMENTO COM AUTOMAÇÃO ROBÓTICA DE PROCESSOS - RPA

Texto base

12

Usando SOAP e REST

Osvaldo Kotaro Takai & Ana Cristina dos Santos

Resumo

Os objetivos desta lição são dois: 1) Aprender a utilizar Web Services com SOAP e 2) Aprender a utilizar o Web Services com REST.

1. Introdução

O que existe de comum hoje em dia são os serviços web para serem consumidos. Esses serviços podem ser acessados utilizando, basicamente, SOAP ou REST.

Assim, o primeiro bot desta lição se concentrará no acesso aos serviços web usando SOAP. Para tanto, o [Serviço Web dos Correios](#) (CORREIOS, 2021) para consultar CEP será utilizado para exemplificar o consumo desse serviço, que é gratuito, pelo bot.

O segundo bot se concentrará no acesso aos serviços web usando REST com o mesmo objetivo de consultar CEP, porém usando os serviços disponibilizados pela [ViaCEP](#).

2. SOAP

A estrutura do bot é apresentada na Figura 1.

Figura 1 - Estrutura do Bot: SOAP



Fonte: do Autor, 2021.

O bot é extremamente simples, haja vista que ele só possui um passo. Entretanto, o processo de preparação é mais denso pois exige o entendimento de como realizar o acesso ao serviço disponibilizado pelos Correios.

2.1. Preparação 1: Web Services

No Manual para Integração via Web Services disponibilizado pelos Correios (CORREIOS, 2021), os serviços web para testes podem ser acessados neste endereço:

<https://apphom.correios.com.br/SigepMasterJPA/AtendeClienteService/AtendeCliente?wsdl>

A página desse endereço no browser será similar ao da Figura 1.

Figura 1 - Página de Serviços Web dos Correios

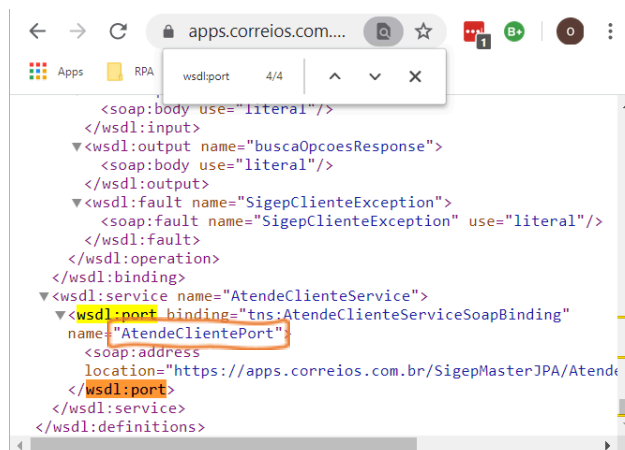
This XML file does not appear to have any style information associated with it. The document tree is shown below:

```
<xsd:definitions xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://cliente.beam.master.sigep.bsb.correios.com.br/" xmlns:tns1="http://schemas.xmlsoap.org/soap/http" name="AtendeClienteService" targetNamespace="http://cliente.beam.master.sigep.bsb.correios.com.br/">
  <xsd:types>
    <xsd:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://cliente.beam.master.sigep.bsb.correios.com.br/" attributeFormDefault="unqualified"
      elementFormDefault="unqualified" targetNamespace="http://cliente.beam.master.sigep.bsb.correios.com.br/">
      <xs:element name="VerificaSeTodosObjetosCancelados" type="tns:VerificaSeTodosObjetosCanceladosResponse"/>
      <xs:element name="VerificaSeTodosObjetosCanceladosResponse" type="tns:VerificaSeTodosObjetosCanceladosResponse"/>
      <xs:element name="atualizaPagamentoNaEntrega" type="tns:atualizaPagamentoNaEntrega"/>
      <xs:element name="atualizaPagamentoNaEntregaResponse" type="tns:atualizaPagamentoNaEntregaResponse"/>
      <xs:element name="atualizaRemessaAgrupada" type="tns:atualizaRemessaAgrupada"/>
      <xs:element name="atualizaRemessaAgrupadaResponse" type="tns:atualizaRemessaAgrupadaResponse"/>
      <xs:element name="bloquearObjeto" type="tns:bloquearObjeto"/>
      <xs:element name="bloquearObjetoResponse" type="tns:bloquearObjetoResponse"/>
      <xs:element name="buscaCliente" type="tns:buscaCliente"/>
      <xs:element name="buscaClienteResponse" type="tns:buscaClienteResponse"/>
      <xs:element name="buscaContrato" type="tns:buscaContrato"/>
      <xs:element name="buscaContratoResponse" type="tns:buscaContratoResponse"/>
      <xs:element name="buscaDataAtual" type="tns:buscaDataAtual"/>
      <xs:element name="buscaDataAtualResponse" type="tns:buscaDataAtualResponse"/>
      <xs:element name="buscaOpcoes" type="tns:buscaOpcoes"/>
      <xs:element name="buscaOpcoesResponse" type="tns:buscaOpcoesResponse"/>
      <xs:element name="buscaPagamentoEntrega" type="tns:buscaPagamentoEntrega"/>
      <xs:element name="buscaPagamentoEntregaResponse" type="tns:buscaPagamentoEntregaResponse"/>
      <xs:element name="buscaServicos" type="tns:buscaServicos"/>
      <xs:element name="buscaServicosAdicionaisAtivos" type="tns:buscaServicosAdicionaisAtivos"/>
      <xs:element name="buscaServicosAdicionaisAtivosResponse" type="tns:buscaServicosAdicionaisAtivosResponse"/>
      <xs:element name="buscaServicosResponse" type="tns:buscaServicosResponse"/>
      <xs:element name="buscaServicosValorDeclarado" type="tns:buscaServicosValorDeclarado"/>
      <xs:element name="buscaServicosValorDeclaradoResponse" type="tns:buscaServicosValorDeclaradoResponse"/>
      <xs:element name="buscaServicosXServicosAdicionais" type="tns:buscaServicosXServicosAdicionais"/>
      <xs:element name="buscaServicosXServicosAdicionaisResponse" type="tns:buscaServicosXServicosAdicionaisResponse"/>
      <xs:element name="buscaTarifaVale" type="tns:buscaTarifaVale"/>
      <xs:element name="buscaTarifaValeResponse" type="tns:buscaTarifaValeResponse"/>
      <xs:element name="calculaTarifaServico" type="tns:calculaTarifaServico"/>
      <xs:element name="calculaTarifaServicoResponse" type="tns:calculaTarifaServicoResponse"/>
      <xs:element name="cancelarObjeto" type="tns:cancelarObjeto"/>
      <xs:element name="cancelarObjetoResponse" type="tns:cancelarObjetoResponse"/>
      <xs:element name="cancelarPedidoScol" type="tns:cancelarPedidoScol"/>
      <xs:element name="cancelarPedidoScolResponse" type="tns:cancelarPedidoScolResponse"/>
      <xs:element name="consultaCEP" type="tns:consultaCEP"/>
      <xs:element name="consultaCEPResponse" type="tns:consultaCEPResponse"/>
    </xsd:schema>
  </xsd:types>
</xsd:definitions>
```

Fonte: do Autor, 2021.

No resultado, pode-se pesquisar pela porta do serviço: **wsdl:port** e verificar que nome dessa porta é: **AtendeClientePort** (Figura 2). Este nome deve ser anotado, pois será utilizado mais tarde na configuração da ação que fará a chamada ao serviço.

Figura 2 - Porta do Serviço consultaCEP



Fonte: do Autor, 2021.

Para visualizar melhor os serviços disponibilizados neste endereço, é interessante instalar o Plugin do Chrome denominado [Boomerang](#) (Figura 3).

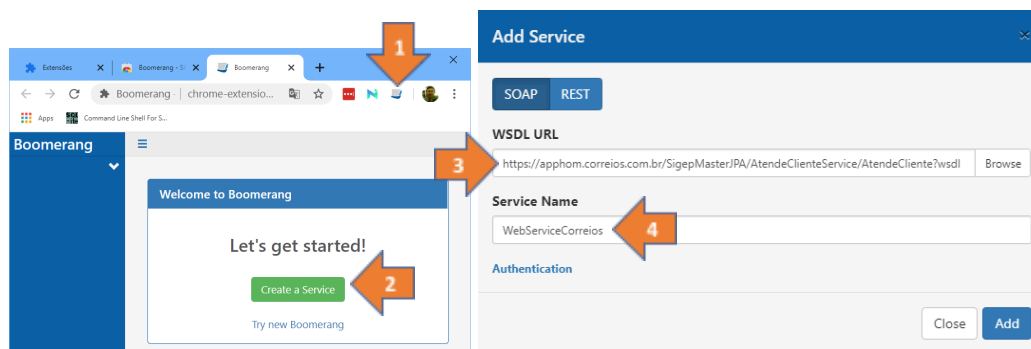
Figura 3 - Plugin Boomerang



Fonte: do Autor, 2021.

Após a instalação do Boomerang e clicando no ícone do Boomerang, pode-se criar um serviço com base no endereço dos serviços web dos Correios (Figura 4). Neste serviço criado, foi atribuído o nome: WebServiceCorreios.

Figura 4 - Criando um serviço usando o Boomerang

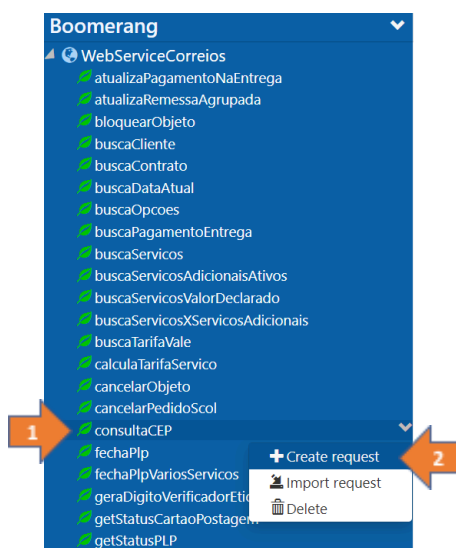


Fonte: do Autor, 2021.

No lado esquerdo irão aparecer todos os serviços disponibilizados pelos Correios. A maioria são acessíveis por assinatura. Nesta lição, será acessado o serviço gratuito consultaCEP:

- 1) Clicar em **consultaCEP**.
- 2) Selecionar **create a request** para criar uma solicitação para este serviço.

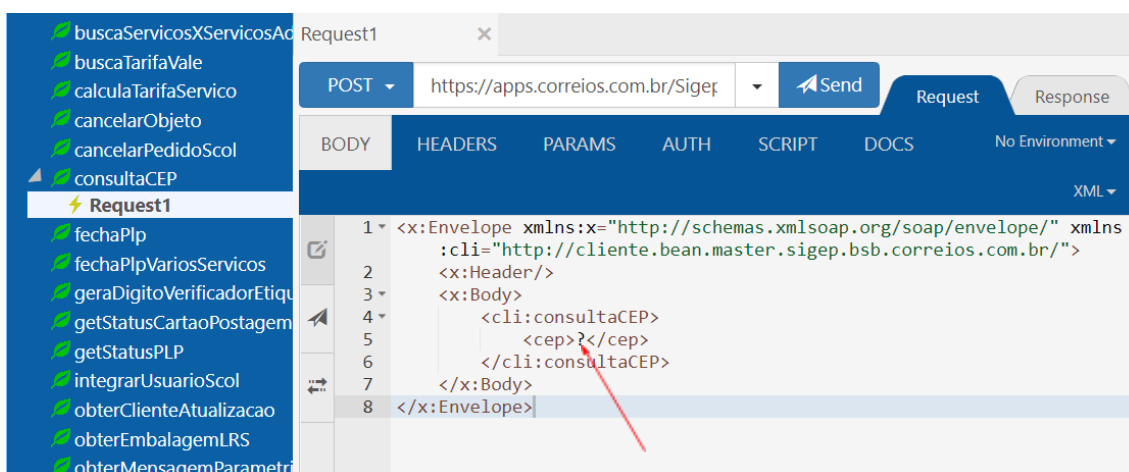
Figura 5 - Criando o request para acessar a operação consultaCEP



Fonte: do Autor, 2021.

Ao criar a requisição, o Boomerang exibe o XML da requisição onde o caractere de interrogação (?) indica o local onde se deve colocar o CEP a ser consultado (Figura 6).

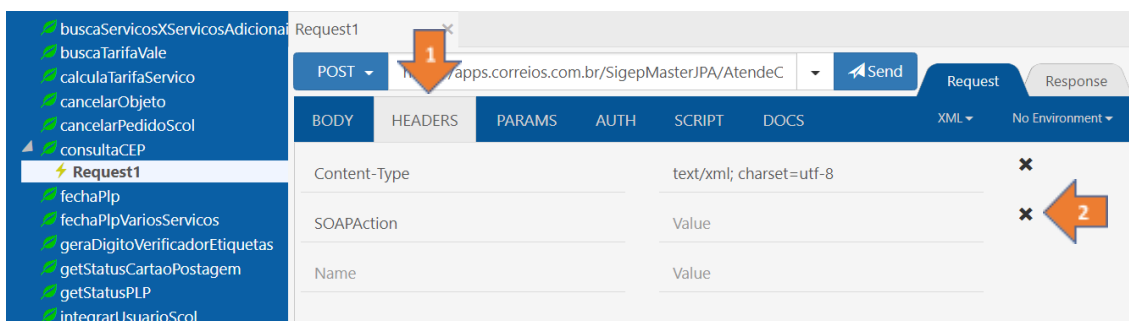
Figura 6 - Request de consulta



Fonte: do Autor, 2021.

Antes de enviar a consulta, é necessário ir em HEADERS (1) e remover a linha SOAPAction (2), pois esta consulta não é desse tipo e causará problemas caso seja mantida (Figura 7).

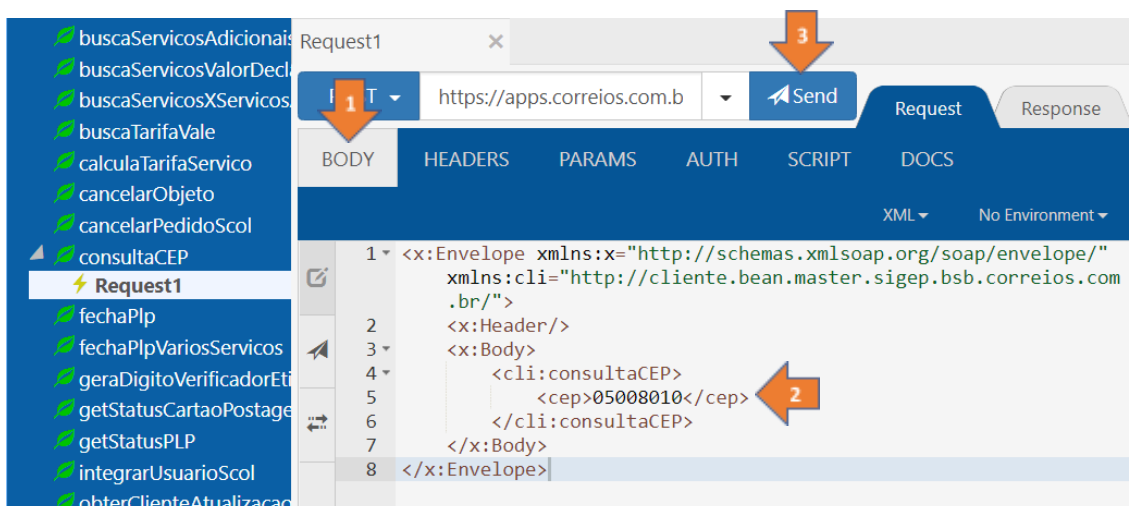
Figura 7 - Remoção da opção SOAPAction



Fonte: do Autor, 2021.

Retornar para BODY (1), substituir a interrogação por um CEP válido (2) e enviar a consulta, Send (3) (Figura 8).

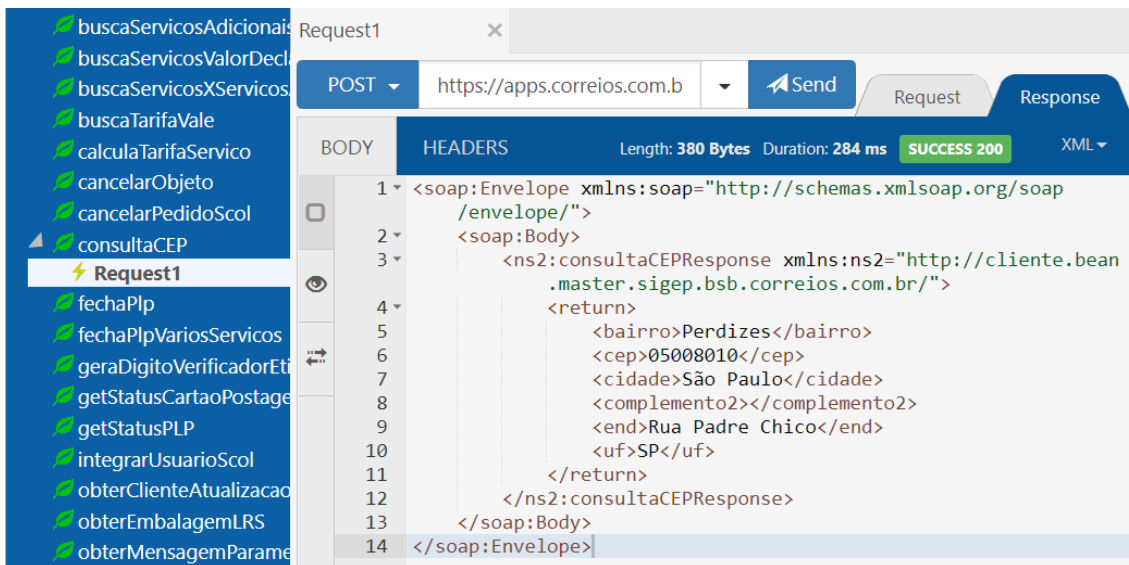
Figura 8 - Enviando a consulta



Fonte: do Autor, 2021.

Pela análise da resposta do serviço, pode-se verificar que a resposta é uma String XML onde cada um dos valores associados ao CEP consultado está entre tags, por exemplo <bairro>Perdizes</bairro> (Figura 9).

Figura 9 - Resposta da consulta ao CEP



Fonte: do Autor, 2021.

2.2. Preparação 2: Consultas XPath

Com base na análise do XML de resposta (Figura 9), verifica-se que é necessário extrair cada elemento retornado na tag <return>. Para descobrir e testar como realizar as extrações, foi utilizado o FreeFormatter e foi copiada a resposta XML na janela **XML input** (Figura 10):

<https://www.freeformatter.com/xpath-tester.html>.

Figura 10- Página do FreeFormatter

XML Input

Option 1: Copy-paste your XML document here

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:consultaCEPResponse xmlns:ns2="http://cliente.bean.master.sigep.bsb.correios.com.br/">
      <return>
        <bairro>Perdizes</bairro>
        <cep>05008010</cep>
        <cidade>São Paulo</cidade>
        <complemento2></complemento2>
        <end>Rua Padre Chico</end>
        <uf>SP</uf>
      </return>
    </ns2:consultaCEPResponse>
  </soap:Body>
</soap:Envelope>
  
```

Option 2: Or upload your XML document

Escolher arquivo Nenhum arquivo selecionado UTF-8

XPath expression

Include the XML item type in output:

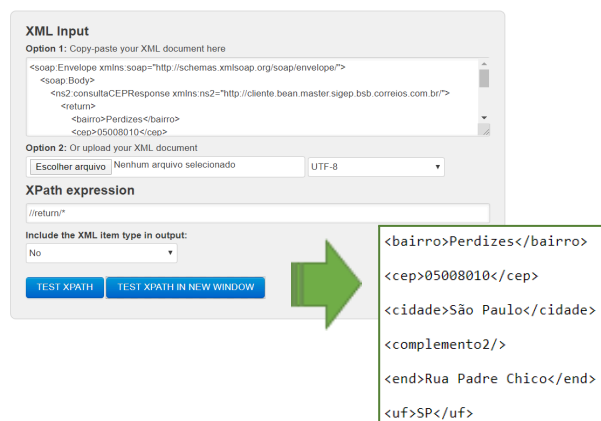
Yes

TEST XPATH TEST XPATH IN NEW WINDOW

Fonte: do Autor, 2021.

No campo XPath expression pode-se entrar com consultas para verificar o resultado. Nesta lição, verificou-se que a consulta: **//return/*** permite obter todas as informações desejadas do CEP consultado (Figura 11). Para se aprofundar sobre consultas XPath pode-se consultar [XPath Introduction](#).

Figura 11 - Consulta XPath: //return/*

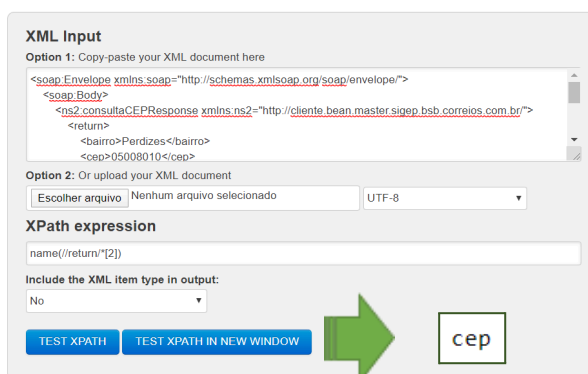


Fonte: do Autor, 2021.

Apenas por curiosidade, pode-se obter cada elemento retornado colocando um número entre colchetes. Por exemplo, para obter o primeiro elemento, bairro, pode-se utilizar a expression: **//return/*[1]**. Esta forma de obter os elementos não será utilizado no bot, pois a ação Loop que será utilizada permite percorrer esses elementos sem precisar de um índice numérico.

Agora, é importante aprender a obter o nome da tag de um elemento. Por exemplo, para obter o nome da tag do segundo elemento, por exemplo, pode-se utilizar: **name(//return/*[2])** (Figura 12)

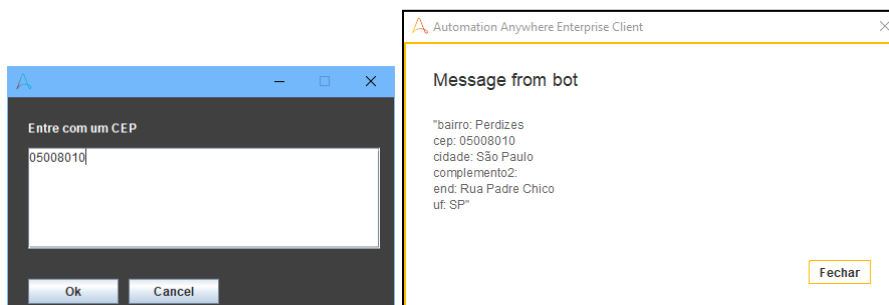
Figura 12 - Obtendo o nome da tag do segundo elemento



Fonte: do Autor, 2021

Agora já se tem todo o conhecimento necessário para construir o bot para consultar o endereço de um dado CEP (Figura 13).

Figura 13 - Telas do bot SOAP



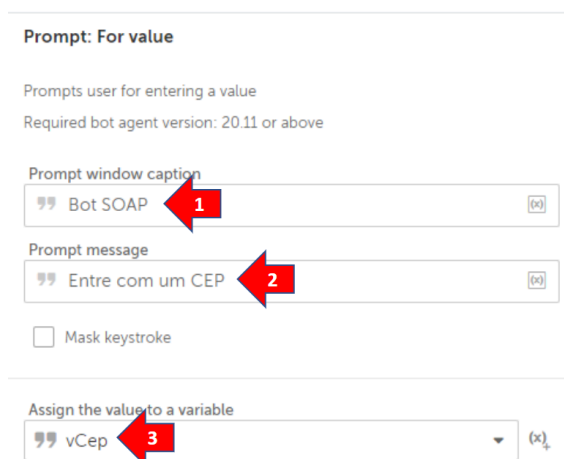
Fonte: do Autor, 2021.

2.3. Prompt: For value

Criar um novo bot e arrastar a ação para que o bot possa solicitar o CEP a ser consultado (Figura 14). Para tanto:

- 1) Atribuir o título da caixa de diálogo.
- 2) Definir a mensagem ao usuário.
- 3) Criar a variável **vCep** e associá-la à ação para que ela possa guardar o CEP digitado pelo usuário.

Figura 14 - Configuração da ação Prompt: For value



Fonte: do Autor, 2021.

2.4. String: Replace

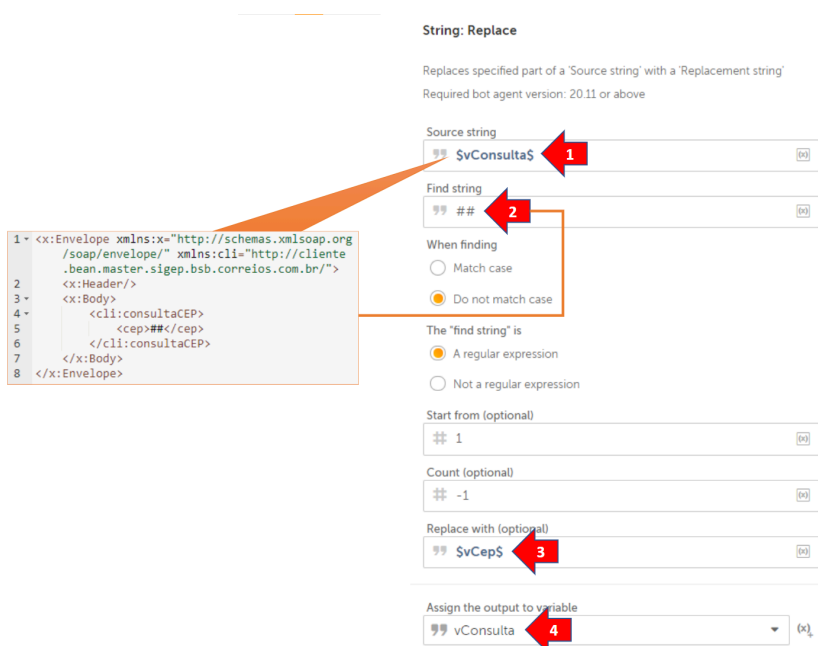
Esta ação coloca o CEP digitado no XML de consulta (Figura 15).

- 1) Criar **vConsulta** e inserir o XML de consulta com o valor do CEP “###”:

```
<x:Envelope xmlns:x="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cli="http://cliente.bean.master.sigep.bsb.correios.com.br/">
  <x:Header/>
  <x:Body>
    <cli:consultaCEP>
      <cep>##</cep>
    </cli:consultaCEP>
  </x:Body>
</x:Envelope>
```

- 2) Indicar o desejo de buscar pela string: **##**.
- 3) Indicar o desejo de trocar a string “**##**” pelo conteúdo da variável **vCep**.
- 4) Colocar o resultado na mesma variável: **vConsulta**.

Figura 15 - Configuração da ação String: Replace



Fonte: do Autor, 2021.

2.5. SOAP Web Service

Esta ação faz a chamada do serviço dos Correios para consultar o CEP (Figura 16).

- 1) Definir o URI de testes do Web Services dos Correios:

```
https://apphom.correios.com.br/SigepMasterJPA/AtendeClien
teService/AtendeCliente?wsdl.
```

- 2) Definir o nome do serviço que será chamado: **AtendeClienteService**.
- 3) Definir a porta de acesso a esse serviço: **AtendeClientePort**.
- 4) Definir a versão do SOAP: **1.1**.
- 5) Indicar a operação que será executada: **consultaCEP**.
- 6) Indicar o XML de requisição que está na variável **vConsulta**.
- 7) Indicar que não há necessidade de autenticação para acessar o serviço.
- 8) Criar a variável **vResposta** e associá-la à ação para que receba a resposta do Web Services dos Correios.

Figura 16 - Configuração da ação SOAP Web Service

The screenshot shows the 'SOAP Web Service' configuration window. It is divided into two main sections: 'Connect to WSDL by' and 'Authentication Mode'. The 'Connect to WSDL by' section has a 'URI' tab selected, showing a text field with the URL 'https://apphom.correios.com.br/SigepMasterJPA/'. Below this is a 'Service' field with 'AtendeClienteService', a 'Port' field with 'AtendeClientePort', and a 'SOAP version' dropdown set to '1.1'. The 'Operation' field is set to 'consultaCEP'. The 'Parameter details' section has a 'Raw data' tab selected, showing a text field with the variable '\$vConsulta\$'. The 'Authentication Mode' section is set to 'No Authentication'. Below this is a 'Client certificate (optional)' section with 'Control Room file', 'Desktop file', and 'Variable' tabs. The 'Custom headers (optional)' section has a table with columns 'Enabled', 'Name', and 'Value', and a button 'Add header'. The 'Output settings' section has a 'Save as XML (optional)' section with a 'Browse...' button, a 'Response Scope (optional)' section with 'Complete response' and 'Selected response' buttons, and an 'Assign the output to a variable' section with a dropdown set to 'vResposta'.

1. URI field: `https://apphom.correios.com.br/SigepMasterJPA/`

2. Service field: `AtendeClienteService`

3. Port field: `AtendeClientePort`

4. SOAP version dropdown: `1.1`

5. Operation field: `consultaCEP`

6. Raw data field: `$vConsulta$`

7. Authentication Mode dropdown: `No Authentication`

8. Assign the output to a variable dropdown: `vResposta`

Fonte: do Autor, 2021.

2.6. XML: Start session

Esta ação cria a sessão para poder consultar a XML de resposta (Figura 17):

- 1) Definir o nome da sessão: **Default**.
- 2) Indicar **vResposta** como sendo a XML que será consultada.

Figura 17 - Configuração da ação XML: Start session

XML: Start session

Start xml session

Required bot agent version: 20.11 or above

Session name

Default

Data Source

☐ File

Control Room file Desktop file Variable

Choose...

☒ Text

\$vResposta\$

Fonte: do Autor, 2021.

2.7. XML: Get multiple nodes

Esta ação define a consulta que será realizada e utilizada pelo Loop da próxima ação (Figura 18):

- 1) Definir o nome da sessão: **Default**.
- 2) Definir a expressão XPath de consulta ao XML: **//return/***.
- 3) Indicar qual resultado se deseja obter; neste caso: **Xpath expression**. Isso permite que sejam obtidos tanto o nome da tag quanto o valor associado.

Figura 18 - Configuração da ação XML: Get multiple nodes

XML: Get multiple nodes

Fetch value from multiple xml nodes

Required bot agent version: 20.11 or above

Session name

Default

XPath Expression

//return/*

For example //bookstore/book

Get each node

☐ Text value

☒ Xpath expression

☐ Specific attribute name

Fonte: do Autor, 2021.

2.8. Loop

Estação permite percorrer cada resultado obtido na ação anterior (Figura 19):

- 1) Definir que o tipo do Loop é o de percorrer cada nodo do XML.
- 2) Definir o nome da sessão: **Default**.
- 3) Criar a variável **vNodo** e associá-la à ação para que ela guarde o nodo de cada iteração.

Figura 19 - Configuração da ação Loop

Loop

Repeats the actions in a loop until a break

Required bot agent version: 20.11 or above

Loop Type

☒ Iterator

Iterator

XML

For each Node in a XML Dataset

Iterator for each node in XML Dataset

Session name

Default

Assign the current row to this variable

vNodo

Fonte: do Autor, 2021.

2.9. XML: Execute XPath function

Para o nodo corrente, obtém o nome da tag correspondente (Figura 20):

- 1) Definir o nome da sessão: **Default**.
- 2) Definir a expressão XPath que permite obter o nome do nodo: **name(\$vNodo\$)**.
- 3) Criar a variável **vNomeDoNo** e associá-la à ação para que ela contenha o nome do nodo da iteração.

Figura 20 - Configuração da ação XML: Execute XPath function

XML: Execute XPath function

Executes the XPath function on the XML
Required bot agent version: 20.11 or above

Session name
Default

XPath expression
name(\$vNode\$)
e.g. /bookstore/book/title

Assign the output to variable
vNomeDoNo

Fonte: do Autor, 2021.

2.9. XML: XML: Get single node (Legacy)

Aqui, foram utilizadas as ações para manipular exceções (Figura 21):

- 1) Na linha 9, tenta-se obter o valor do nodo.
 - a) Definir o nome da sessão: **Default**.
 - b) Definir a expressão XPath para obter o valor do nodo: **\$vNode\$/text()**. Como esta expressão pode gerar erros quando um tag não tiver valor, esta ação foi colocada dentro do **Error handler: Try**.
 - c) Criar a variável **vValorDoNo** e associe-a à ação para guardar o valor do nodo.

Figura 21 - Configuração da ação XML: XML: Get single node (Legacy)

XML: Get single node (Legacy)

Fetch value of a specific node from the xml
Required bot agent version: 20.11 or above

Session name
Default

XPath expression
\$vNode\$/text()
e.g. /bookstore/book/title

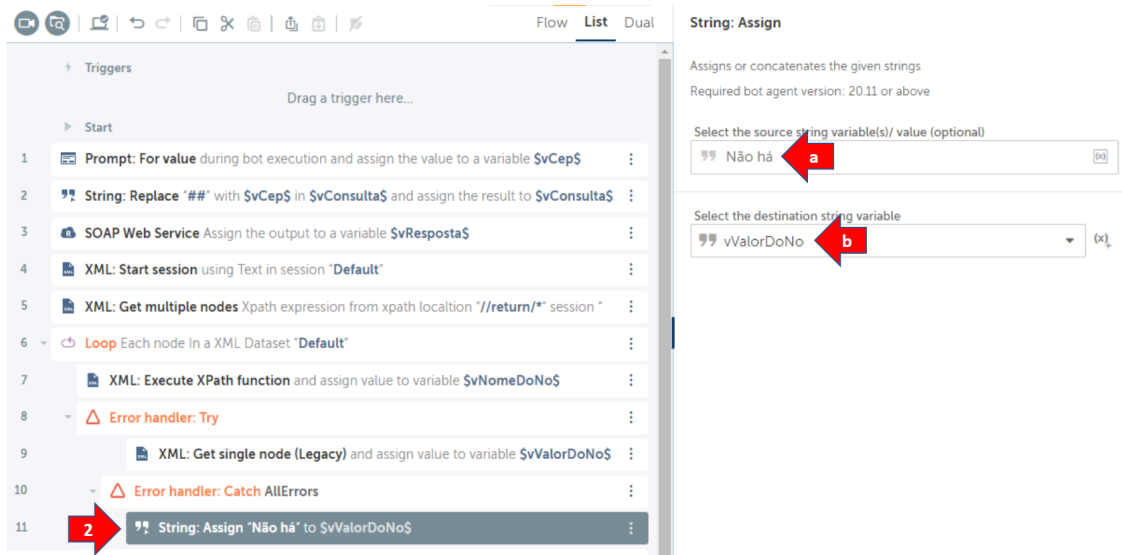
Attribute (optional)

Assign the output to variable
vValorDoNo

Fonte: do Autor, 2021.

- 2) Se der erro, então atribuir “Não há” à variável **vValorDoNo** (Figura 22):
 - a) Definir o valor “Não há”.
 - b) Indicar que a variável **vValorDoNo** receberá o valor “Não há”.

Figura 22 - Configuração da ação XML: XML: Get single node (Legacy)



Fonte: do Autor, 2021.

2.10. String: Assign

Coloca na variável **vNoMaisValor** a concatenação dos pares **vNomeDoNo** e **vValorDoNo** (Figura 23):

- 1) Colocar inicialmente o acumulador **vNoMaisValor**.
- 2) Seguido pelo **vNomeDoNo** convertido para maiúsculas.
- 3) Seguido pelo caractere dois pontos “:”.
- 4) Seguido pelo **vValorDoNo**.
- 5) E, por fim, seguido por **Enter**.
- 6) Colocar o acumulador **vNoMaisValor** para receber tudo.

Figura 23 - Configuração da ação String: Assign

String: Assign

Assigns or concatenates the given strings

Required bot agent version: 20.11 or above

Select the source string variable or value (optional)

1 2 3 4 5

” \$vNoMaisValor\$\$vNomeDoNo.String:uppercase\$ \$vValorDoNo\$\$String:Enter\$ ”

Select the destination string variable

6

” vNoMaisValor ”

Fonte: do Autor, 2021.

2.11. XML: End Session

Esta ação finaliza a sessão Default do XML.

2.12. Message box

Esta ação exibe os detalhes do logradouro associado ao CEP (Figura 24).

Figura 24 - Configuração da ação Message box

Message box

Displays a message box

Required bot agent version: 20.11 or above

Enter the message box window title

” Automation Anywhere Enterprise Client ”

Enter the message to display

” \$vNoMaisValor\$ ”

Scrollbar after lines

30

Fonte: do Autor, 2021.

Ao executar este bot, o resultado deve ser bem similar ao da Figura 13.

3. Bot REST

O objetivo desta lição é aprender a construir um bot que acesse Web Services usando [REST](#).

Para tanto, será criado um Bot que consultará o endereço de um dado CEP nos Serviços Web disponibilizados pela [ViaCEP](#) (Figura 25)

Figura 25 - Estrutura do Bot REST



Fonte: do Autor, 2021

3.1. Preparação

Para consultar o endereço de um CEP no ViaCEP, basta digitar na linha de endereços do navegador, o seguinte:

```
https://viacep.com.br/ws/05008010/piped/
```

Onde 05008010 é o CEP que se deseja consultar.

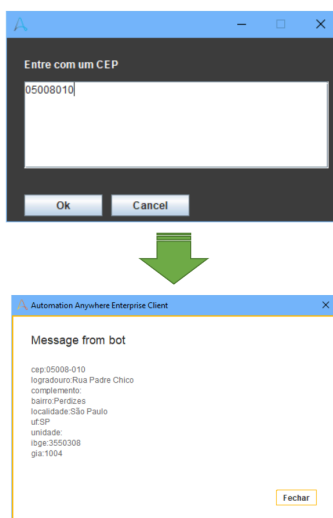
O resultado será a linha:

```
cep:05008-010|logradouro:Rua Padre  
Chico|complemento:|bairro:Perdizes|localidade:São  
Paulo|uf:SP|unidade:|ibge:3550308|gia:1004
```

As barras verticais (pipe) apareceram porque foi solicitado, na linha de consultas, que o resultado fosse “piped”.

É só isso. Agora, pode-se criar um Bot que consulte o endereço de um CEP usando o Serviço Web da ViaCEP, e exiba a resposta numa Caixa de Mensagens (Figura 26).

Figura 26 - Telas do Bot REST



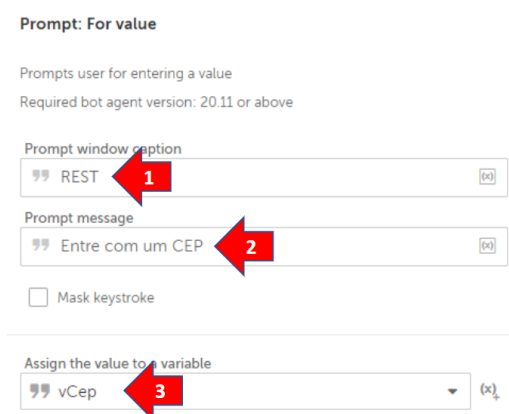
Fonte: do Autor, 2021.

3.2. Prompt: For value

Esta ação permite que o usuário entre com o número do CEP para obter o logradouro correspondente (Figura 27):

- 1) Atribuir o título da caixa de diálogo.
- 2) Definir a mensagem ao usuário.
- 3) Definir onde o CEP digitado será guardado.

Figura 27 - Configuração da ação Prompt: For value



Fonte: do Autor, 2021.

3.2. String: Replace

Esta ação coloca o CEP digitado no XML de consulta (Figura 28):

- 1) Criar a variável **vConsulta** e inserir o XML de consulta com o valor do CEP **“##”**.
- 2) Indicar o desejo de buscar pela string: **##**.
- 3) Indicar o desejo de trocar a string **“##”** pelo conteúdo da variável **vCep**.
- 4) Colocar o resultado na mesma variável: **vConsulta**.

Figura 28 - Configuração da ação String: Replace

String: Replace

Replaces specified part of a 'Source string' with a 'Replacement string'

Source string: 1

Find string: 2

When finding:

☐ Match case

☒ Do not match case

The "find string" is:

☐ A regular expression

☒ Not a regular expression

Start from (optional):

Count (optional):

Replace with: 3

Assign the output to variable: 4

<https://viacep.com.br/ws/##/piped/>

Fonte: do Autor, 2021.

3.3. REST Web Services: Get method

Esta ação realiza a consulta de CEP via REST (Figura 29):

- 1) Definir o URI do Web Services REST do ViaCEP: **\$vConsulta\$**.
- 2) Indicar que não há necessidade de se autenticar para acessar este serviço.
- 3) Criar a variável dicionário e associá-la à ação para receber a resposta da consulta.
- 4) Importante observar as orientações de que a resposta estará na chave **“body”** do dicionário.

Figura 29 - Configuração da ação REST Web Services: Get method

Fonte: do Autor, 2021.

3.4. Dictionary: Get

Esta ação atribui à variável **vSaida** o resultado da consulta que se encontra na chave “Body” do dicionário **vResposta** (Figura 30):

- 1) Criar e indicar a variável dicionário: **vResposta**.
- 2) Indicar a chave: **Body**.
- 3) Criar a variável **vSaida** e associá-la à ação para que ela contenha somente o corpo necessário da resposta.

Figura 29 - Configuração da ação Dictionary: Get

Dictionary: Get

Returns the entry associated with the key in the Dictionary.
Required bot agent version: 20.11 or above

Dictionary variable
vResposta 1

Key
Body 2

Assign the output to variable
vSaida 3

Fonte: do Autor, 2021.

3.5. String: Replace

Como a **vSaida** contém “pipes” separando os valores, será necessário trocar os “pipes” por ENTER (Figura 30):

- 1) Indicar a string fonte: **vSaida**.
- 2) Indicar a string que se deseja encontrar para trocá-la: “|”.
- 3) Indicar por qual string o “Pipe” deverá ser trocado: **\$String:Enter\$**.
- 4) Indicar que a string resultante será guardada na string original: **vSaida**.

Figura 30 - Configuração da ação String: Replace

String: Replace

Replaces specified part of a 'Source string' with a 'Replacement string'

Source string
vSaida\$ 1

Find string
| 2

When finding
☐ Match case
☒ Do not match case

The "find string" is
☐ A regular expression
☒ Not a regular expression

Start from (optional)
1

Count (optional)
-1

Replace with
\$String:Enter\$ 3

Assign the output to variable
vSaida 4

Fonte: do Autor, 2021.

3.6. Message box

Esta ação exibe os detalhes do logradouro associado ao CEP (Figura 31).

Figura 31 - Configuração da ação Message box

The screenshot shows the configuration interface for a 'Message box' action. It includes the following fields:

- Message box**: Displays a message box. Required bot agent version: 20.11 or above.
- Enter the message box window title**: A text box containing 'Automation Anywhere Enterprise Client'.
- Enter the message to display**: A text box containing 'SvNoMaisValor\$'.
- Scrollbar after lines**: A numeric input field set to '30'.

Fonte: do Autor, 2021.

Ao executar este bot, o resultado deve ser bem similar ao da Figura 26.

4. Considerações finais

Nesta parte foram desenvolvidos dois bots. O primeiro para construir um bot que utilizou o SOAP para acessar o Web Services dos Correios e o segundo para acessar o Web Services do ViaCEP usando REST.

Com isso, finaliza-se as lições desta disciplina e espera-se que os alunos estejam, agora, aptos e preparados para prosseguir seus estudos de forma autônoma no mundo do RPA.

Referências

AUTOMATION ANYWHERE IQ BOT. **Como usar IQ Bot**; 2020. Disponível em <<https://docs.automationanywhere.com/pt-BR/bundle/iq-bot-v6.5/page/iq-bot/topics/iq-bot/user/overview.html>>, acesso em 15/08/2021.

AUTOMATION ANYWHERE AUTOMATION 360. **Using dictionary variable for email properties.** Disponível em <<https://docs.automationanywhere.com/bundle/enterprise-v2019/page/enterprise-cloud/topics/aae-client/bot-creator/commands/cloud-using-email-properties.html>>, acesso em 21 de julho de 2021.

AUTOMATION ANYWHERE COMMUNITY EDITION. **Formulário para obtenção de acesso à versão Community Edition do Automation Anywhere Gratuita.** São José – EUA. Disponível em <<https://www.automationanywhere.com/products/enterprise/community-edition>>, acesso em 21 de junho de 2021.

AUTOMATION ANYWHERE UNIVERSITY. **Introdução ao Automation Anywhere.** São José – EUA. Disponível em <<https://apeople.automationanywhere.com/s/getting-started>>, acesso em 21 de junho de 2021a.

AUTOMATION ANYWHERE UNIVERSITY. **Trilhas de Aprendizagem.** São José – EUA. Disponível em: <<https://university.automationanywhere.com/training/rpa-learning-trails/>>, acesso em 21 de junho de 2021b.

AUTOMATION ANYWHERE UNIVERSITY. **Email server setting.** Disponível em <<https://docs.automationanywhere.com/bundle/enterprise-v2019/page/enterprise-cloud/topics/aae-client/bot-creator/commands/cloud-configuring-mail-server.html>>, acesso em 23 de julho de 2021c.

BANIN, S. L. **Python 3: conceitos e aplicações: uma abordagem didática.** São Paulo; Érica, 2018. ISBN 978-85-365-3025-3. <https://integrada.minhabiblioteca.com.br/#/books/9788536530253/>

CHICONI, N. O que é ASCII, UNICODE e UTF-8. CCM, 2020. Disponível em <<https://br.ccm.net/faq/9956-o-que-e-ascii-unicode-e-utf-8>>. Acesso em 21 de julho de 2021.

CHANDRA, R. V.; VARANASI, B. S. **Python Requests Essentials: Learn how to integrate your applications seamlessly with web services using Python Requests**; Packt Publishing, 2015.

CORREIOS; Manual para Integração via Web Services. Diretoria de Negócios dos Correios – Departamento de Canais Digitais. 30/06/2021. Disponível em <[http://www.corporativo.correios.com.br/encomendas/sigepweb/doc/Manual de Implementacao do Web Service SIGEP WEB.pdf](http://www.corporativo.correios.com.br/encomendas/sigepweb/doc/Manual_de_Implementacao_do_Web_Service_SIGEP_WEB.pdf)>. Acesso em 05 de setembro de 2021.

DIGICERT. THE ULTIMATE GUIDE: What is SSL, TLS and HTTPS. Disponível em <<https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>>, acesso em 23 de julho de 2021.

ELMAN, J.; LAVIN, M. **Django Essencial - Usando REST, websockets e Backbone**. 1.ed. São Paulo: Novatec, 2015.

Elmasri, R.; Navathe, S. B. [Trad.]. **Sistemas de bancos de dados**. Traduzido do original: **FUNDAMENTALS OF DATABASE SYSTEMS**. São Paulo: Pearson(Addison Wesley), 2005. 724 p. ISBN: 85-88639-17-3.

GOOGLE. Ajuda do Administrador do Google Workspace: Controle o acesso a apps menos seguros. Disponível em <<https://support.google.com/a/answer/6260879?hl=pt-BR>>. Acesso em 21 de julho de 2021.

JARMUL, K.; LAWSON, R. Python Web Scraping. 2nd. Birmingham: Packt Publishing, 2017.

LOPES, M. D. e LIMA, W. R. **Análise do Índice de Massa Corporal de funcionários de uma instituição de ensino superior**; EFDeportes.com, Revista Digital. Buenos Aires - Año 18 - N° 181 - Junio de 2013. Disponível em <<https://www.efdeportes.com/efd181/analise-do-indice-de-massa-corporal-de-funcionarios.htm>>, acesso em 21 de julho de 2021.

MCINGVALE, FRANK. **All about Python and Unicode**; trad. Menezes, Nilo; PythonBrasil, 2007. Disponível em <<https://wiki.python.org.br/TudoSobrePythoneUnicode>>, acesso em 29 de julho de 2021.

MICROSOFT. O que são IMAP e POP? Disponível em <<https://support.microsoft.com/pt-br/office/o-que-s%C3%A3o-imap-e-pop-ca2c5799-49f9-4079-aefe-ddca85d5b1c9>>, acesso em 23 de julho de 2021.

PYTHON BRASIL. **Instalando o Python 3 no Windows**. Disponível em <<https://python.org.br/instalacao-windows/>>, acesso em 24 de julho de 2021.

NOTEPAD++, **What is Notepad++**. Disponível em <<https://notepad-plus-plus.org/>>, acesso em 28 de junho de 2021.

VIACEP. Consulte CEPs de todo o Brasil. Disponível em: <<https://viacep.com.br/>>. Acesso em 05 de setembro de 2021.

WDG AUTOMATION – AN IBM COMPANY. **7 pilares essenciais para projetos de RPA bem-sucedidos**. São Paulo: Newsletter WDG. Disponível em: <<https://www.wdgautomation.com/7-pilares-essenciais-para-projetos-de-rpa-bem-sucedidos/>>, acesso em 21 de junho de 2021.