

# Séries Temporais do BACEN

## Objetivo

- Obter dados diretamente do site do **Sistema Gerenciador de Séries Temporais (SGS)** do Banco Central do Brasil ([www.bacen.gov.br](http://www.bacen.gov.br)).
- Criar/Atualizar planilhas e arquivos com o conteúdo da série
- Utilizar arquivos de configuração como listas: *list*

## Descrição do Problema

O Sistema de Séries Temporais do BACEM permite a automatização do acesso às séries temporais através de um Webservice.

Para utilizar esta funcionalidade é necessária a instalação de alguns *packages* do R.

# HTTP e HTML

## HTTP: *Hypertext Transfer Protocol*

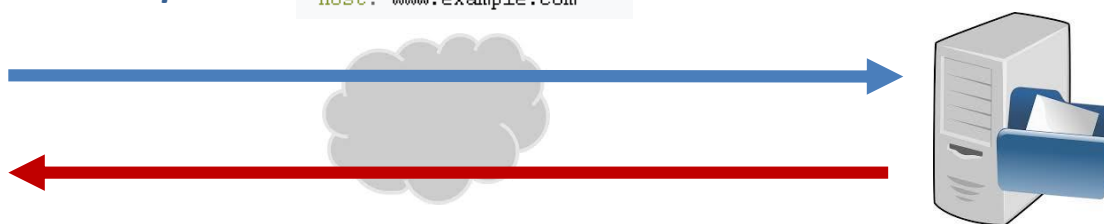
- Protocolo de comunicação cliente-servidor
- Dados no formato texto e hipertexto (HTML: Hypertext Markup Language = texto + links)
- *Request-Response*: 1) Abre a conexão; 2) Envia *request* (*GET* ou *POST*); 3) Recebe *response*; 4) Fecha a conexão

**Cliente: Browser**

**Request:**

```
GET /index.html HTTP/1.1
Host: www.example.com
```

**Servidor: Web Server**



**Response:**

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close
```

Header

**HTML: tags e atributos**

```
<tag nome_atributo="valor_atributo">
  (...) texto (...)
</tag>
```



```
<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  <h1> Hello World!</h1>
  <a href="http://www.example.com">link to example</a>
</body>
</html>
```

Body: HTML

# XML (*Extensible Markup Language*)

## XML (*Extensible Markup Language*)

- *Markup Language* texto utiliza *TAGs* e *Atributos* de TAGs para definir estruturas complexas de dados no formato 100% texto.
- Estruturas de dados podem ser definidas através de esquemas “*schemas*” para validação do formato e dos dados

### XML Schema: Definição da estrutura

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="https://www.w3schools.com"
xmlns="https://www.w3schools.com"
elementFormDefault="qualified">

  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="message" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```



### XML

```
<?xml version="1.0"?>

<note
  xmlns="https://www.w3schools.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://www.w3schools.com note.xsd">
  <to>José</to>
  <from>Maria</from>
  <heading>Lembrete</heading>
  <message>Não se esqueça de se lembrar! </message>
</note>
```

root element: *note*

element: *to*

element: *from*

element: *heading*

element: *message*

*note*

*to: José*

*from: Maria*

*heading: Lembrete*

*message: Não se esquecer de se lembrar*

# Webservice SOAP = HTTP + XML

## SOAP (*Simple Object Access Protocol*)

Protocolo de transmissão de estruturas de dados complexas utilizando XML e baseado no protocolo HTTP como camada de comunicação

## WSDL (*Web Services Description Language*)

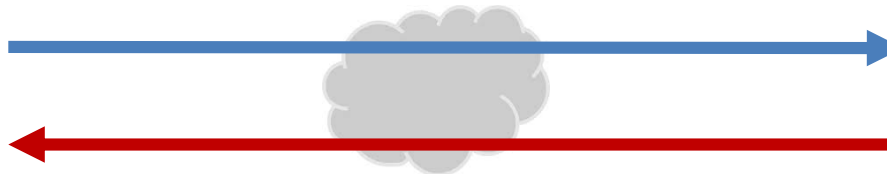
Linguagem para descrever quais funcionalidades estarão disponíveis em um webservice

**Cliente:** *Aplicação*

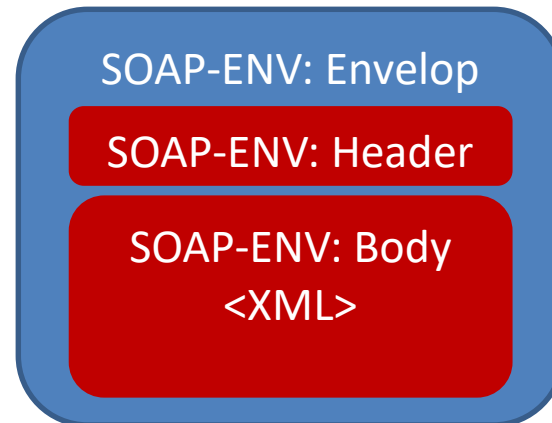


*Request padronizado: HTTP*

**Servidor:** *Webservice*



*Response: HTML e XML*



# BACEN: Webservice de Acesso a Séries Históricas



SGS - Sistema Gerenciador de Séries Temporais - v2.1  
Módulo público

<https://www3.bcb.gov.br/sgspub/localizarseries/localizarSeries.do?method=prepararTelaLocalizarSeries>

## Serviços automatizados (webservices)

- O SGS - Sistema Gerenciador de Séries Temporais disponibiliza serviços de consulta utilizando a tecnologia de WebServices. As definições (wsdl) dos serviços estão na seguinte URL:  
[FachadaWSSGS?wsdl](#)  
Para consumir o webservice do SGS, é necessário importar a cadeia de certificados do Banco Central, que pode ser encontrada na página <http://www.bcb.gov.br/?CERTDIG>.
- Segue-se abaixo uma breve descrição dos serviços disponibilizados:
  - getUltimoValorVO** - Recupera o último valor de uma determinada série e retorna um objeto do tipo WSSerieVO.  
Parâmetros:  
  long codigoSerie - Código da série.  
Retorno:  
  WSSerieVO - Objeto série.
  - getUltimoValorXML** - Recupera o último valor de uma determinada série e retorna o resultado em formato XML.  
Parâmetros:  
  long codigoSerie - Código da série.  
Retorno:  
  String - String contendo o resultado da consulta em formato XML.
  - getValor** - Recupera o valor de uma série em uma determinada data (dd/MM/aaaa).  
Parâmetros:  
  long codigoSerie - Código da série.  
  String data - String contendo a data (dd/MM/aaaa) do valor a ser pesquisado.  
Retorno:  
  BigDecimal - Objeto contendo o valor.
  - getValorEspecial** - Recupera o valor de uma série especial em um período.  
Parâmetros:  
  long codigoSerie - Código da série.  
  String data - String contendo a data (dd/MM/aaaa) inicial.  
  String dataFim - String contendo a data (dd/MM/aaaa) final.  
Retorno:  
  BigDecimal - Objeto contendo o valor.
  - getValoresSeriesXML** - Recupera os valores de uma ou mais séries dentro de um determinado período. O resultado da consulta é devolvido ao cliente em formato XML.  
Parâmetros:  
  long[] codigosSeries - Lista(array) dos códigos das séries.  
  String dataInicio - String contendo a data (dd/MM/aaaa) inicial.  
  String dataFim - String contendo a data (dd/MM/aaaa) final.  
Retorno:  
  String - String contendo o resultado da consulta em formato XML.
  - getValoresSeriesVO** - Recupera os valores de uma ou mais séries dentro de um determinado período e retorna o resultado em forma de Array de objetos do tipo WSSerieVO.  
Parâmetros:  
  long[] codigosSeries - Lista(array) dos códigos das séries.  
  String dataInicio - String contendo a data (dd/MM/aaaa) inicial.  
  String dataFim - String contendo a data (dd/MM/aaaa) final.  
Retorno:  
  WSSerieVO - Lista(array) de objeto série.

# BACEN: Webservice de Acesso a Séries Históricas

**Objetivo:** Automatizar a consulta de series históricas no Sistema Gerenciador de Séries Históricas do BACEN através do webservice disponibilizado

Função definida na WSDL: **getValoresSeriesXML**

(...)

**getValoresSeriesXML** - Recupera os valores de uma ou mais séries dentro de um determinado período. O resultado da consulta é devolvido ao cliente em formato XML.

Parâmetros:

long[] codigosSeries - Lista(array) dos códigos das séries.

String dataInicio - String contendo a data (dd/MM/aaaa) inicial.

String dataFim - String contendo a data (dd/MM/aaaa) final.

Retorno:

String - String contendo o resultado da consulta em formato XML.

(...)

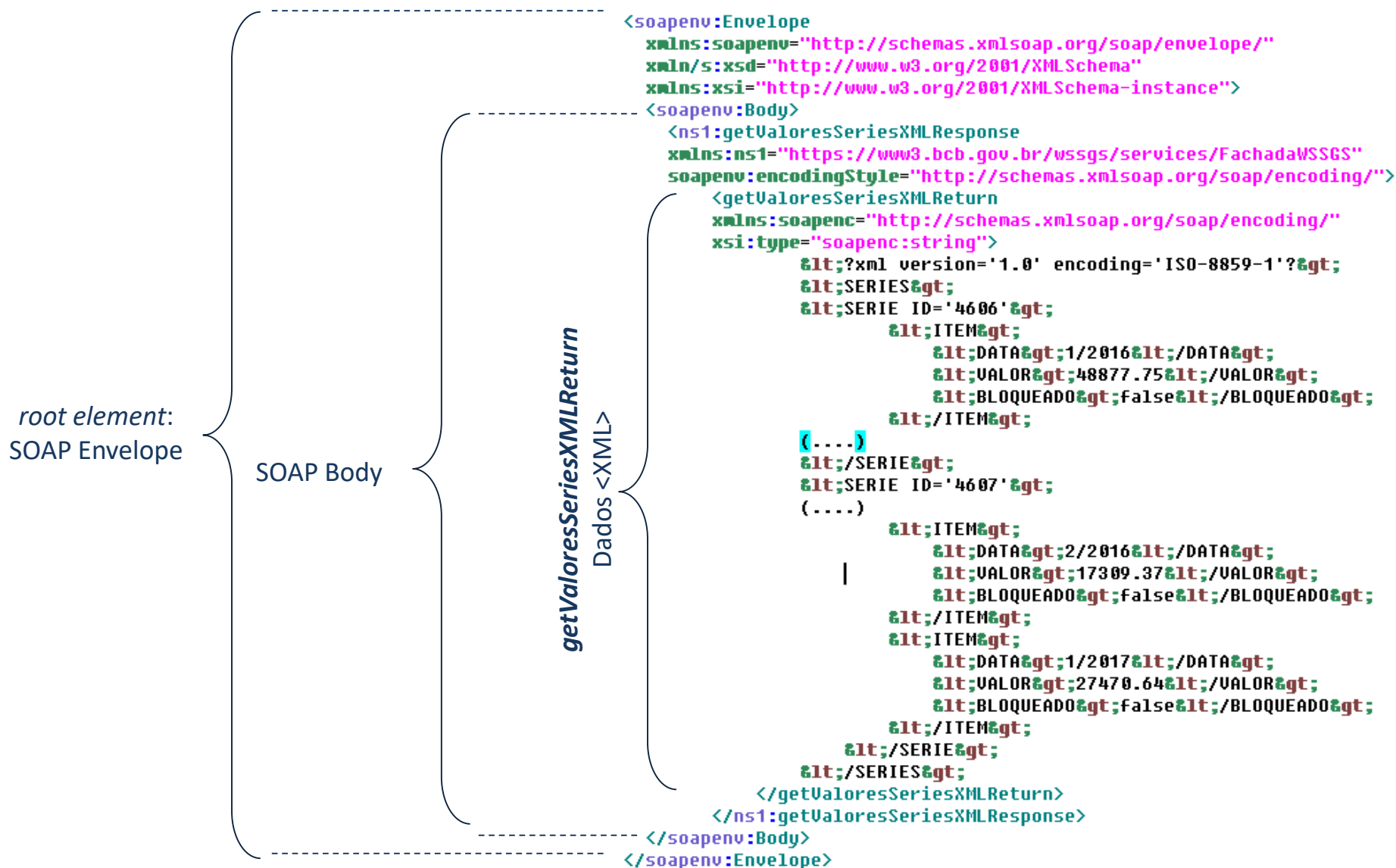
**Exemplo:** Códigos: 4606 e 4607; Data Inicial: 01/01/2016; Data Final: 01/01/2017

Código XML para o *request* via HTTP

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getValoresSeriesXML xmlns="https://www3.bcb.gov.br/wssgs/services/FachadaWSSGS">
      <codigosSeries>
        <number>4606</number>
        <number>4607</number>
      </codigosSeries>
      <dataInicio> 01/01/2016 </dataInicio>
      <dataFim> 01/01/2017 </dataFim>
    </getValoresSeriesXML>
  </soapenv:Body>
</soapenv:Envelope>
```

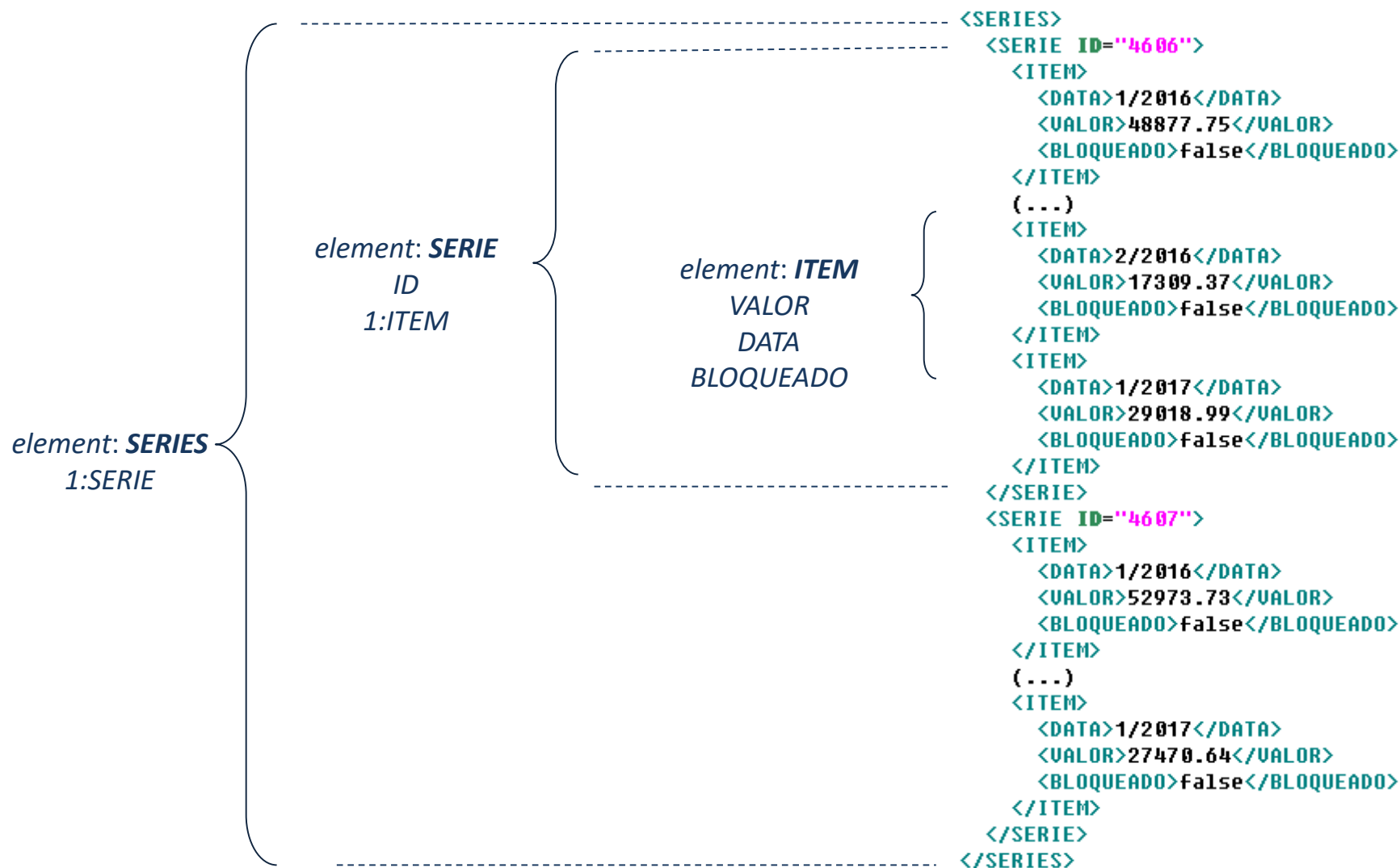
# BACEN: Webservice de Acesso a Séries Históricas

Código XML para o *reponse* via HTTP



# BACEN: Webservice de Acesso a Séries Históricas

Código XML element *getValoresSeriesXMLReturn*



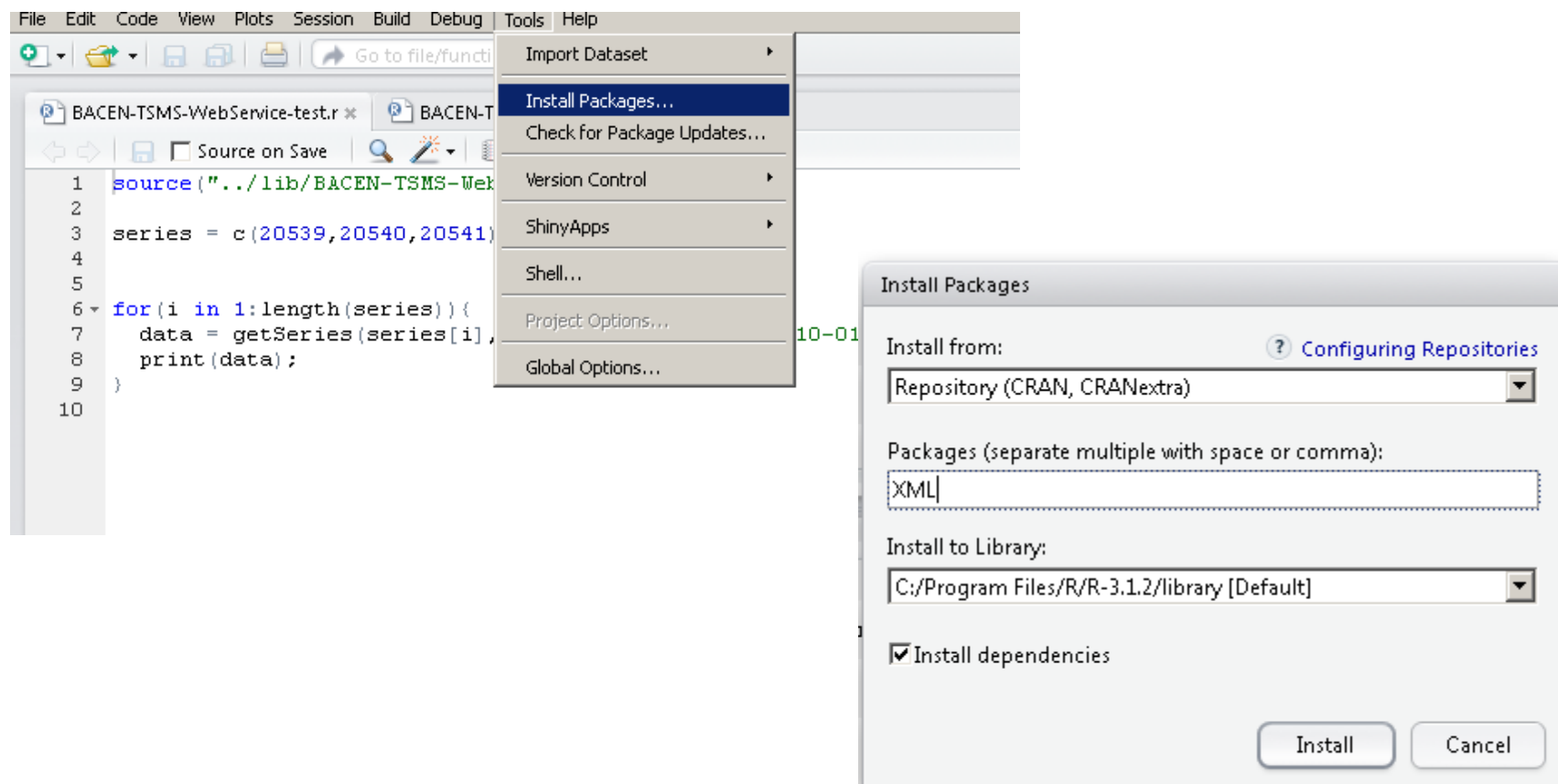


# Séries Temporais do BACEN

## Preparando o R:

Instalar os packages XML, httr e RCurl

Disponíveis em: <https://cran.r-project.org/>



The screenshot shows the RStudio interface. The 'Tools' menu is open, and 'Install Packages...' is selected. The 'Install Packages' dialog box is open, showing the following fields:

- Install from:** Repository (CRAN, CRANextra)
- Packages (separate multiple with space or comma):** XML
- Install to Library:** C:/Program Files/R/R-3.1.2/library [Default]
- ☒ Install dependencies

The 'Install' button is highlighted.

# Séries Temporais do BACEN

Código: BACEN-TSMS-WebService-Explain.r

1. Preparação do cabeçalho e do Corpo com o *request* no formato XML de acordo com o WSDL

```
## Biblioteca para baixar series do BACEN Time Series Management System
library(RCurl);
library(httr);
library(XML);

## URL do Webservice
url = "https://www3.bcb.gov.br/wssgs/services/FachadaWSSGS?method=getValoresSeriesXML";

## Data para input no 'getValoresSeriesXML'
startDateStr = "01/01/2016";
endDateStr = "01/01/2017";

## Séries de input
codes = c(4606,4607);

## Prepara o corpo da mensagem de request "body"
soapenv = '<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getValoresSeriesXML xmlns="https://www3.bcb.gov.br/wssgs/services/FachadaWSSGS">
      <codigosSeries> %s </codigosSeries>
      <dataInicio> %s </dataInicio>
      <dataFim> %s </dataFim>
    </getValoresSeriesXML>
  </soapenv:Body>
</soapenv:Envelope>';

## Corpo do Request
body = sprintf(soapenv,paste(sprintf("<number>%d</number>",codes),collapse=""),startDateStr,endDateStr);
```

# Séries Temporais do BACEN

**Código:** *BACEN-TSMS-WebService-Explain.r*

## 2. Execução do *request* HTTP via POST

```
## POST request
```

```
response = POST(url, body = body,
  add_headers("Content-Type"="text/xml; charset=utf-8",
    "soapAction"="https://www3.bcb.gov.br/wsags/services/FachadaWSSGS/getValoresSeriesXML",
    "Accept-encoding"="zip"));
```

## 3. Tratamento (*Parser*) do *response* com as funções:

*xmlParse(...)*: Parser do conteúdo texto para uma estrutura XML

*xmlRoot (...)*: Extraí o elemento *root*

*xmlValeu (...)*: Extraí conteúdo de um elemento xml

```
## -- Response
```

```
## Envelope com o conteúdo do SOAP
```

```
xmlSOAPEnvelope = xmlRoot(xmlParse(response, asText=TRUE));
```

```
## Valor do Node "getValoresSeriesXMLReturn" = código xml com o resultado da query
```

```
xmlData = xmlRoot(xmlParse(xmlValue(xmlSOAPEnvelope[["Body"]][["getValoresSeriesXMLResponse"]][["getValoresSeriesXMLReturn"]], asText=TRUE));
```

# Séries Temporais do BACEN

## Preparando o R:

Copiar o arquivo `BACEN-TSMS-webService.r` que implementa a função:

```
TSMGetSeries = function(codes,
  startDate = as.Date("2016-01-01"), endDate = Sys.Date(),
  output = c("data.frame", "list", "xml"),
  parseDates = TRUE,
  url = "https://www3.bcb.gov.br/achadaWSSGS?method=getValoresSeriesXML",
  ssl.verifypeer = FALSE)
```

# Séries Temporais do BACEN

## Exemplo de Script

```
source("../lib/BACEN-TSMS-WebService.r");

## Séries a serem coletadas
series = c(20539,20540,20541);

## Modo 01: Várias séries de uma vez

## Chamada de datas como strings
## data = TSMSGetSeries(series,startDate="01/01/2012",endDate="01/01/2014");

## Chamada com datas como datas
## data = TSMSGetSeries(series,startDate=as.Date("2012-01-01"),endDate=as.Date("2014-01-01"));

data = TSMSGetSeries(series);
print(data);

## Modo 02: Uma série de cada vez

for(i in 1:length(series)){

  ## Chamada de datas como strings
  ## data = TSMSGetSeries(series[i],startDate="01/01/2012",endDate="01/01/2014");

  ## Chamada com datas como datas
  ## data = TSMSGetSeries(series[i],startDate=as.Date("2012-01-01"),endDate=as.Date("2014-01-01"));

  ## Chamada com valores default para datas
  data = TSMSGetSeries(series[i]);

  print(data);
}
```

# Séries Temporais do BACEN

## Exercício 01

Desenvolver um script que carrega de um arquivo de configuração com uma lista de séries definidas pelos atributos:

Nome da séries

Data de Início

Data de Fim

Nome do *Workbook*

Nome de *Worksheet*

Baixa os valores do BACEM e atualiza as planilha com as séries.

## Exemplo de dados para o arquivo de configuração:

Nome: Transações correntes - mensal - saldo

Id: 22701,

Data de início: 2010-01-01

Data de Final: 2016-01-01

workbook: ../database/Series-test.xlsx

worksheet: saldo

Nome: Transações correntes - mensal - receitas

Id: 22702,

Data de início: 2010-01-01

Data de Final: 2016-01-01

workbook: ../database/Series-test.xlsx

worksheet: receitas

# Séries Temporais do BACEN

## Exercício 02

Desenvolver a função `BACEN_TSMStoExcelFixHeader`.

**Objetivo:** Baixar as séries do Banco Central e atualizar planilhas Excel com os dados

**Parâmetros:**

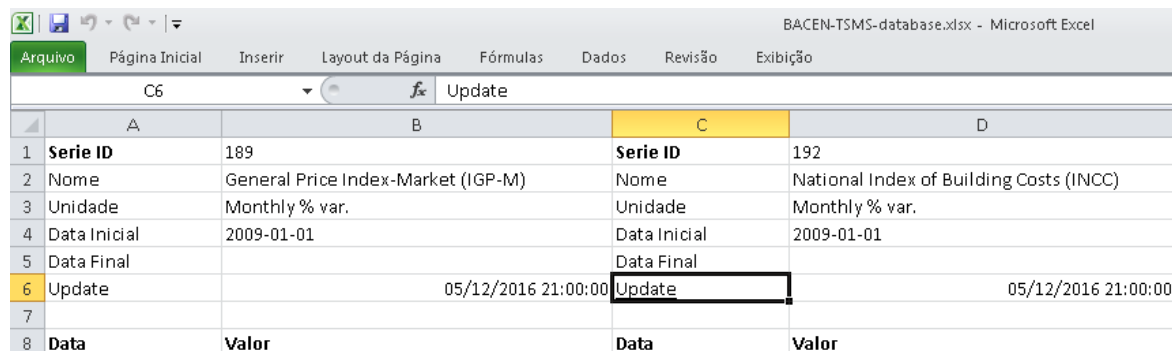
`workbook`: path do arquivo Excel que será atualizado com os dados baixados

`url` (opcional) = <https://www3.bcb.gov.br/sgspub/JSP/sgsgeral/FachadaWSSGS.wsdI>

**Retorno:** Número de séries atualizadas

**Detalhes:**

- A planilha terá um cabeçalho predefinido conforme exemplo
- Verificar na célula A1 se o valor é “Serie ID” se for, ler o ID da séries na célula B1 e baixar a série. Repetir este procedimento a cada duas células até que a célula esteja vazia ou o valor não seja “Serie ID”.
- Ler os valores “Date Inicial” e “Data Final”. Se o valor “Data Final” for vazio considerar hoje
- Atualizar o campo: “Update” com a data da última atualização.



	A	B	C	D
1	Serie ID	189	Serie ID	192
2	Nome	General Price Index-Market (IGP-M)	Nome	National Index of Building Costs (INCC)
3	Unidade	Monthly % var.	Unidade	Monthly % var.
4	Data Inicial	2009-01-01	Data Inicial	2009-01-01
5	Data Final		Data Final	
6	Update	05/12/2016 21:00:00	Update	05/12/2016 21:00:00
7				
8	Data	Valor	Data	Valor

# Séries Temporais do BACEN

## Exercício 03

Desenvolver a função `BACEN_TSMStoExcel`.

**Objetivo:** Baixar as séries do Banco Central e atualizar planilhas Excel com os dados

**Parâmetros:**

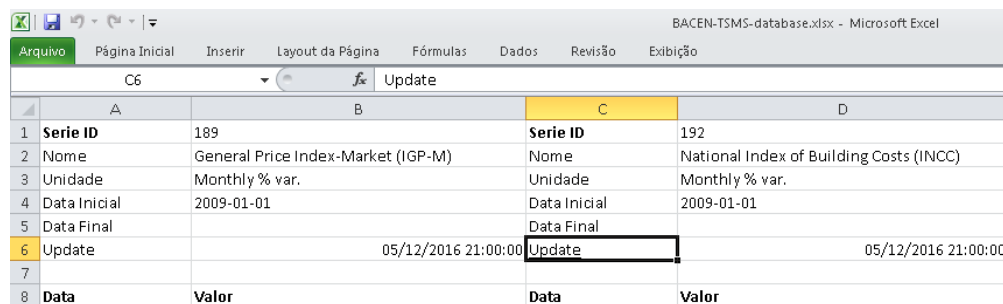
`workbook`: path do arquivo Excel que será atualizado com os dados baixados

`url` (opcional) = <https://www3.bcb.gov.br/sgspub/JSP/sgsgeral/FachadaWSSGS.wsdI>

**Retorno:** Número de séries atualizadas

**Detalhes:**

- A planilha terá um cabeçalho com campos predefinido conforme exemplo **em qualquer ordem**, a menos do “Serie ID” que sempre estará no canto superior direito.
- Verificar na célula A1 se o valor é “Serie ID” se for, ler o ID da séries na célula B1 e baixar a série. Repetir este procedimento a cada duas células até que a célula esteja vazia ou o valor não seja “Serie ID”.
- Ler os valores “Date Inicial” e “Data Final”. Se o valor “Data Final” for vazio considerar hoje.
- Lê os dados de cabeçalho até encontrar uma linha em branco
- Atualizar o campo: “Update” com a data da última atualização.



	A	B	C	D
1	Serie ID	189	Serie ID	192
2	Nome	General Price Index-Market (IGP-M)	Nome	National Index of Building Costs (INCC)
3	Unidade	Monthly % var.	Unidade	Monthly % var.
4	Data Inicial	2009-01-01	Data Inicial	2009-01-01
5	Data Final		Data Final	
6	Update	05/12/2016 21:00:00	Update	05/12/2016 21:00:00
7				
8	Data	Valor	Data	Valor