

DCC917A – TÓPICOS ESPECIAIS III: DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

AULA 10

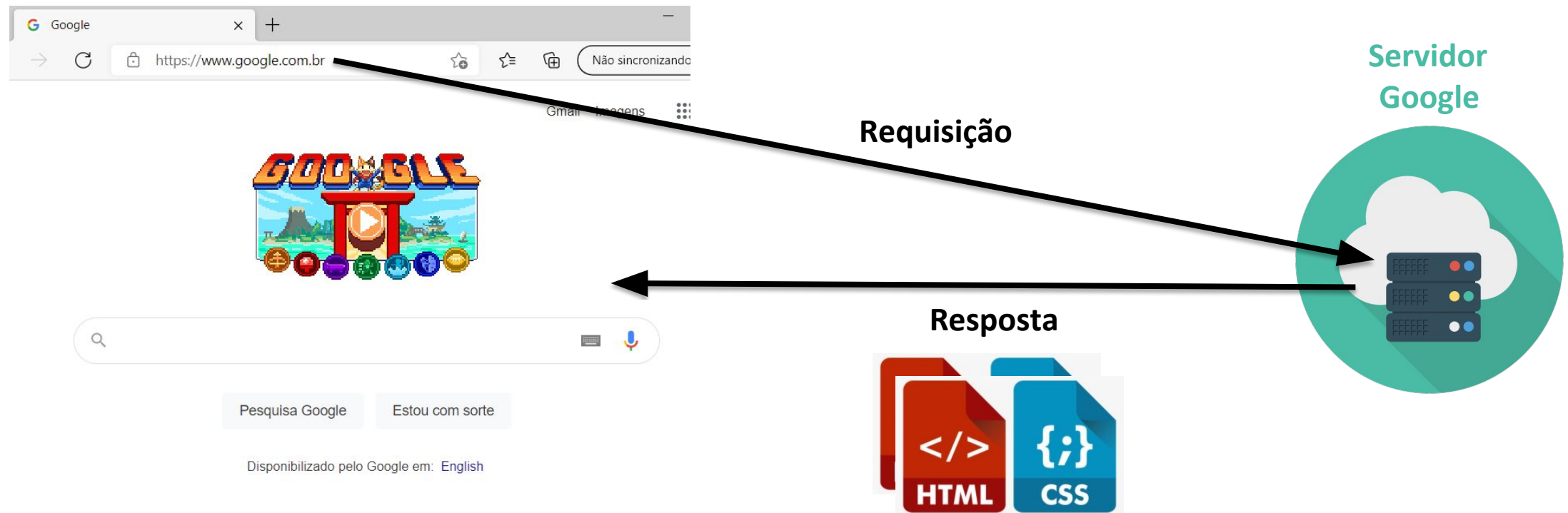
Carlos Bruno Oliveira Lopes

Engenheiro de Computação

Mestre em Ciência da Computação

Web servisse (consumo de serviços)

- O consumo de serviços baseia-se em requisições:



Web servisse (consumo de serviços)

Requisições HTTP

- HTTP (Hypertext Transfer Protocol)
 - Meio de comunicação padronizada entre navegadores e servidores
- Tipos de requisições (Principais)
 - **Get**: recuperar dados no servidor
 - **Post**: criar um dado novo no servidor
 - **Put**: Atualizar dados no servidor
 - **Delete**: Deletar dados no servidor



Usados em App para
acesso a serviços web

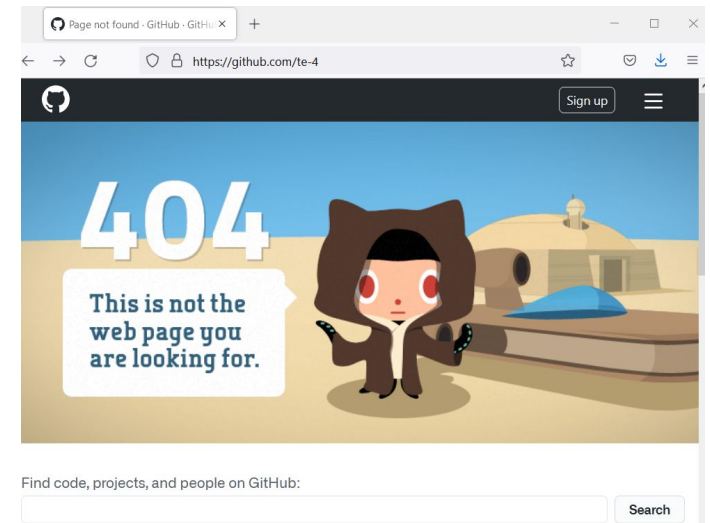
Web servisse (consumo de serviços)

Código de status

- 404: URL não encontrada
- 200: ok (requisição correta)
- 501: Bad Gateway (servidor com problema)

Não há a necessidade de mostrar esse códigos para o usuário.

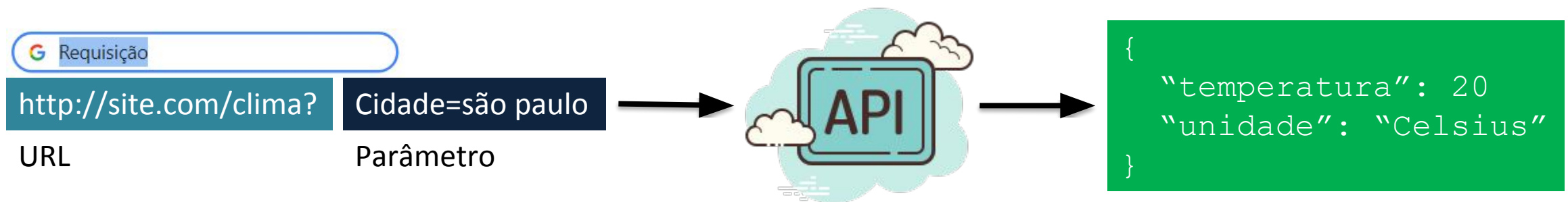
- Precisamos apenas dizer se o recurso esta disponível ou não.
- E possivelmente exibir uma mensagem de disponibilidade.



Web servisse (consumo de serviços)

API

- API (Application Programming Interface) em português “Interface de Programação de Aplicações”
 - A API em que receberá as requisições e responderá as mesmas via HTTP
 - Portanto, utilizaremos a API para realizar comunicação, ou seja, os recursos disponibilizados na web serão acessados por meio de API;



Web servisse (consumo de serviços)

Tipos de retornos API

– XML (eXtensible Markup Language)

- Linguagem de marcação usado para padronizar a comunicação web entre diferentes tecnologias e linguagens de programação

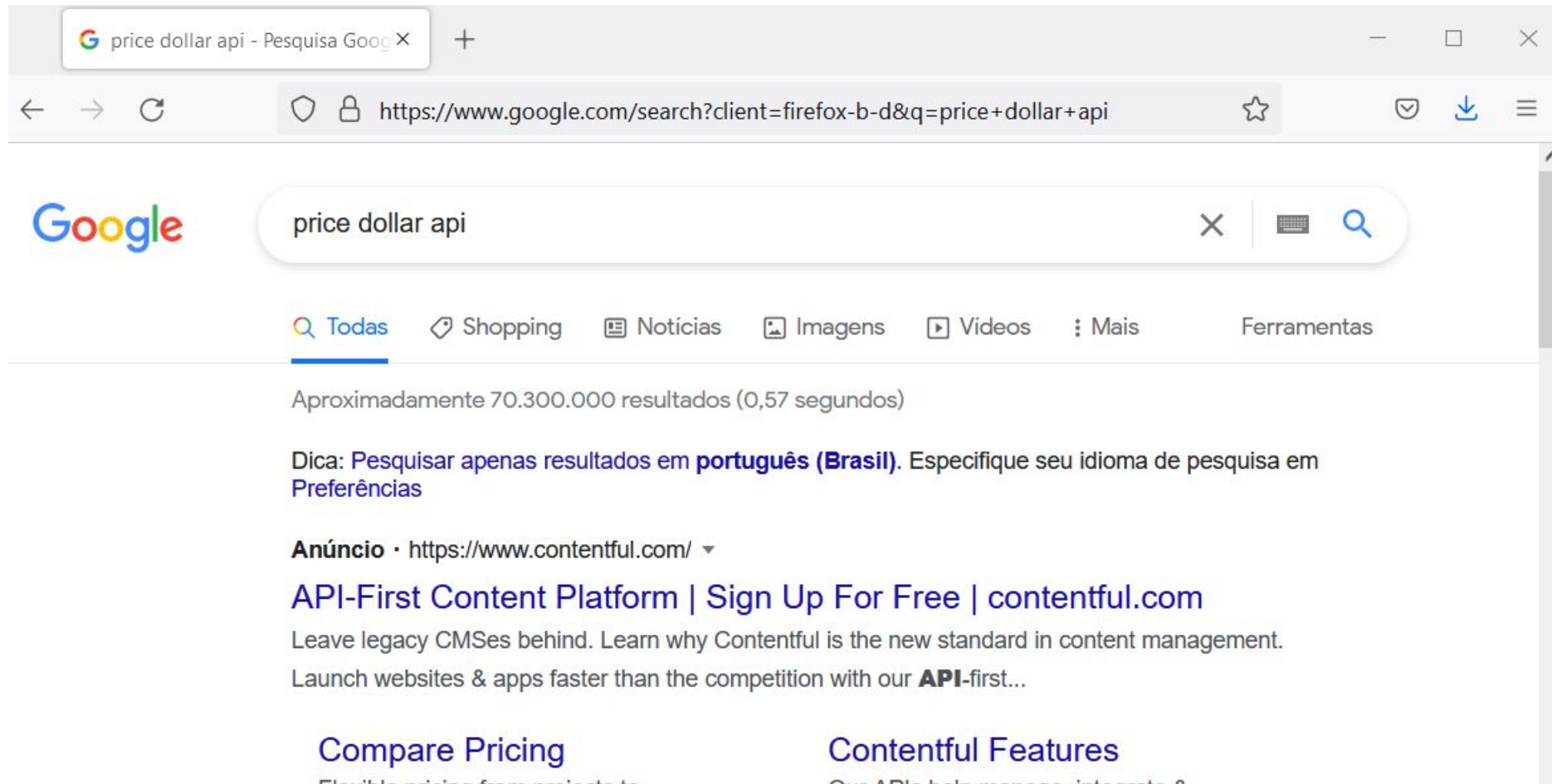
```
<resultado>  
  <temperatura>20</temperatura>  
  <unidade>Celsius</unidade>  
</resultado>
```

– Json (JavaScript Object Notation)

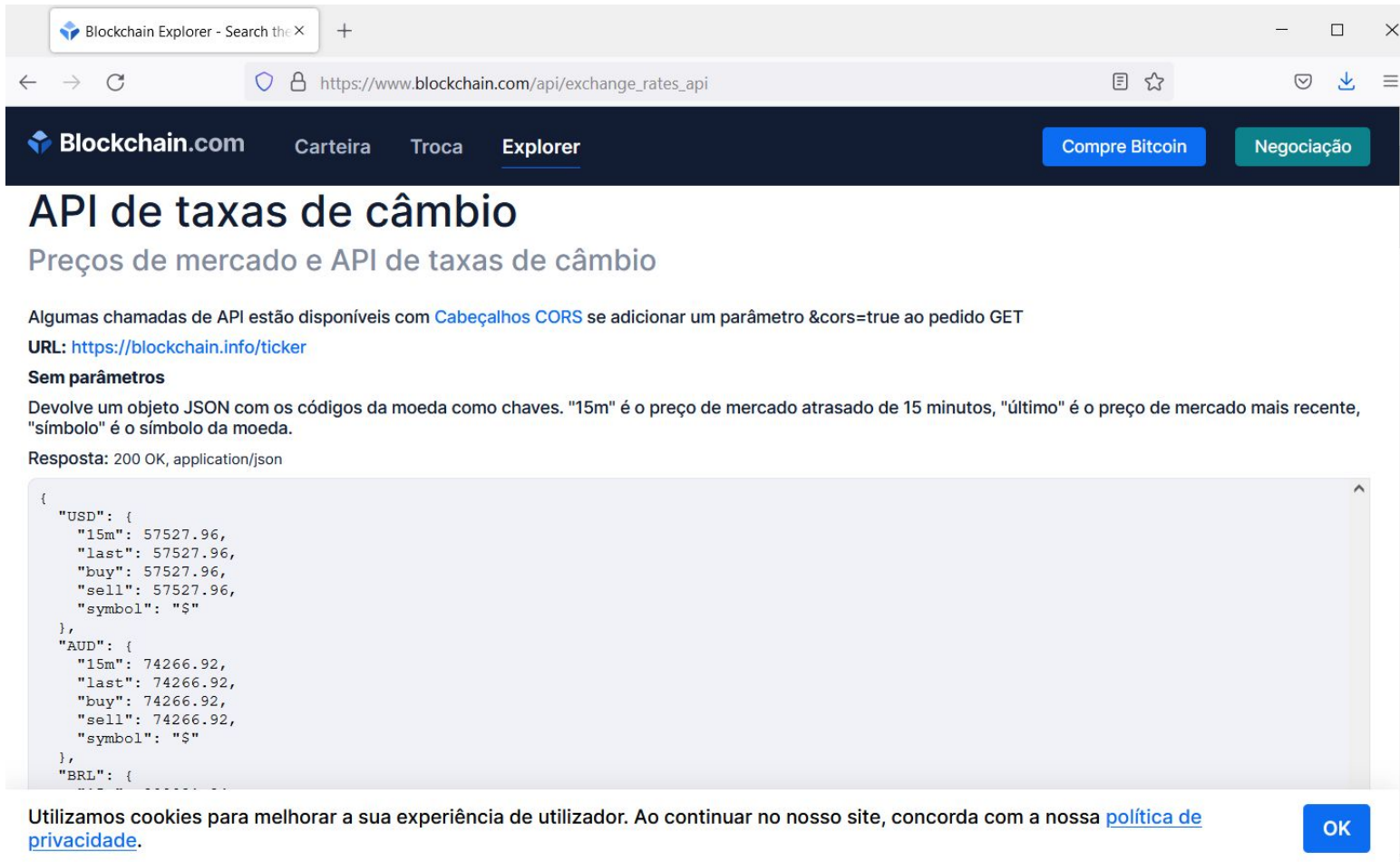
- é um formato compacto, de padrão aberto independente, de troca de dados simples e rápida (parsing) entre sistemas.

```
{  
  "id":1,  
  "nome":"Alexandre Gama",  
  "endereco":"R. Qualquer"  
}
```

Web servisse (consumo de serviços)



Web servisse (consumo de serviços)



The screenshot shows a web browser window with the address bar displaying `https://www.blockchain.com/api/exchange_rates_api`. The page title is "API de taxas de câmbio" (Exchange Rate API). Below the title, it says "Preços de mercado e API de taxas de câmbio". A note mentions that some API calls are available with CORS headers by adding `&cors=true` to the GET request. The URL `https://blockchain.info/ticker` is provided. It states that no parameters are needed. The response is described as a 200 OK with `application/json` content type. A JSON response is shown in a code block, containing data for USD, AUD, and BRL. At the bottom, there is a cookie consent banner.

Blockchain Explorer - Search the X

https://www.blockchain.com/api/exchange_rates_api

Blockchain.com Carteira Troca Explorer Compre Bitcoin Negociação

API de taxas de câmbio

Preços de mercado e API de taxas de câmbio

Algumas chamadas de API estão disponíveis com **Cabeçalhos CORS** se adicionar um parâmetro `&cors=true` ao pedido GET

URL: <https://blockchain.info/ticker>

Sem parâmetros

Devolve um objeto JSON com os códigos da moeda como chaves. "15m" é o preço de mercado atrasado de 15 minutos, "último" é o preço de mercado mais recente, "símbolo" é o símbolo da moeda.

Resposta: 200 OK, application/json

```
{
  "USD": {
    "15m": 57527.96,
    "last": 57527.96,
    "buy": 57527.96,
    "sell": 57527.96,
    "symbol": "$"
  },
  "AUD": {
    "15m": 74266.92,
    "last": 74266.92,
    "buy": 74266.92,
    "sell": 74266.92,
    "symbol": "$"
  },
  "BRL": {
```

Utilizamos cookies para melhorar a sua experiência de utilizador. Ao continuar no nosso site, concorda com a nossa [política de privacidade](#). OK

URL: <https://blockchain.info/ticker>

Sem parâmetros

Retorno: Json

Web servisse (consumo de serviços)

The image is a collage of three web service interfaces. On the left is a Blockchain interface with a dark theme, showing a 'Documentation' sidebar with links like 'Getting started', 'Tutorials', and 'Tools and libraries'. The main content area shows a URL and parameters. In the center is the Twitter Developer Platform interface, featuring a 'Documentation' sidebar and a main area with the heading 'API reference'. On the right is the Viacep interface, which has a green header and provides information about a free web service for consulting Brazilian CEPs, including instructions on how to use it and an example URL.

Blockchain

URL: <https://blockchain.info>

Converte o valor x

ex: <https://blockchain.info>

Parâmetros

- **currency** - U
- **value** - Valo

Devolve o valor em

Resposta:

10

Developer Platform

Documentation

Search the docs

Twitter API

Getting started

Tutorials

Tools and libraries

Migrate

[API reference index](#)

API reference

This resource is the Twitter API endpoints from the entire API Reference Index.

Twitter

Tweets

VIACEP

Consulte CEPs de todo o Brasil

Procurando um [webservice](#) gratuito e de alto desempenho para consultar Códigos de Endereçamento Postal (CEP) do Brasil? Utilize o serviço, melhore a qualidade de suas aplicações web e colabore para manter esta base de dados atualizada.

Acessando o webservice de CEP

Para acessar o webservice, um CEP no formato de {8} dígitos deve ser fornecido, por exemplo: "01001000".

Após o CEP, deve ser fornecido o tipo de retorno desejado, que deve ser "json", "xml", "piped" ou "querty".

Exemplo de pesquisa por CEP:

viacep.com.br/ws/01001000/json/

Validação do CEP

Quando consultado um CEP de formato inválido, por exemplo: "950100100" (9 dígitos),

Flutter (Requisições Web)

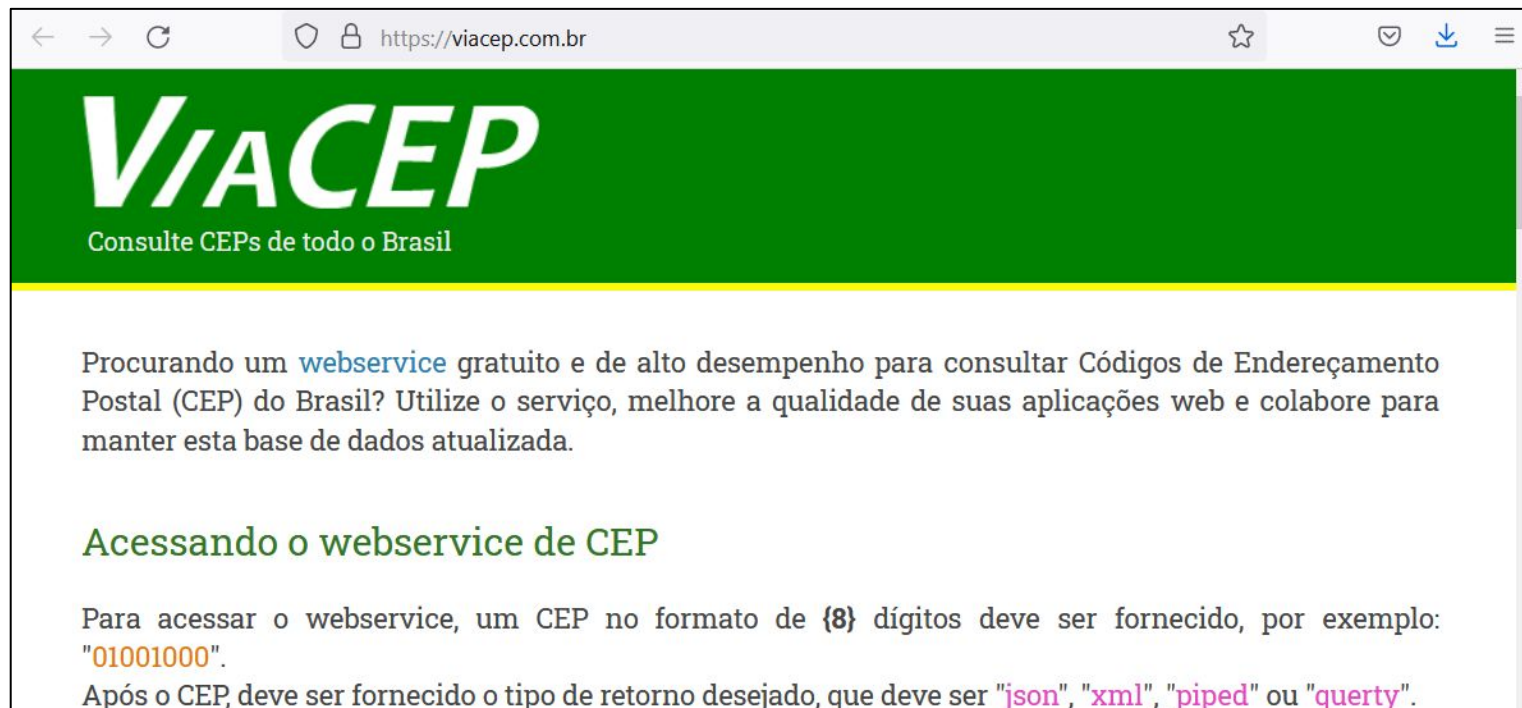
Pacote HTTP (http package)

- Pacote que contém um conjunto de funções e classes de alto nível usada para o fácil consumo de recursos http
- Versão: 0.13.3
- Link: <https://pub.dev/packages/http>

Flutter (Requisições Web)

API para recuperação de endereço usando o CEP

– Link da API: <https://viacep.com.br/>



Flutter (Requisições Web)

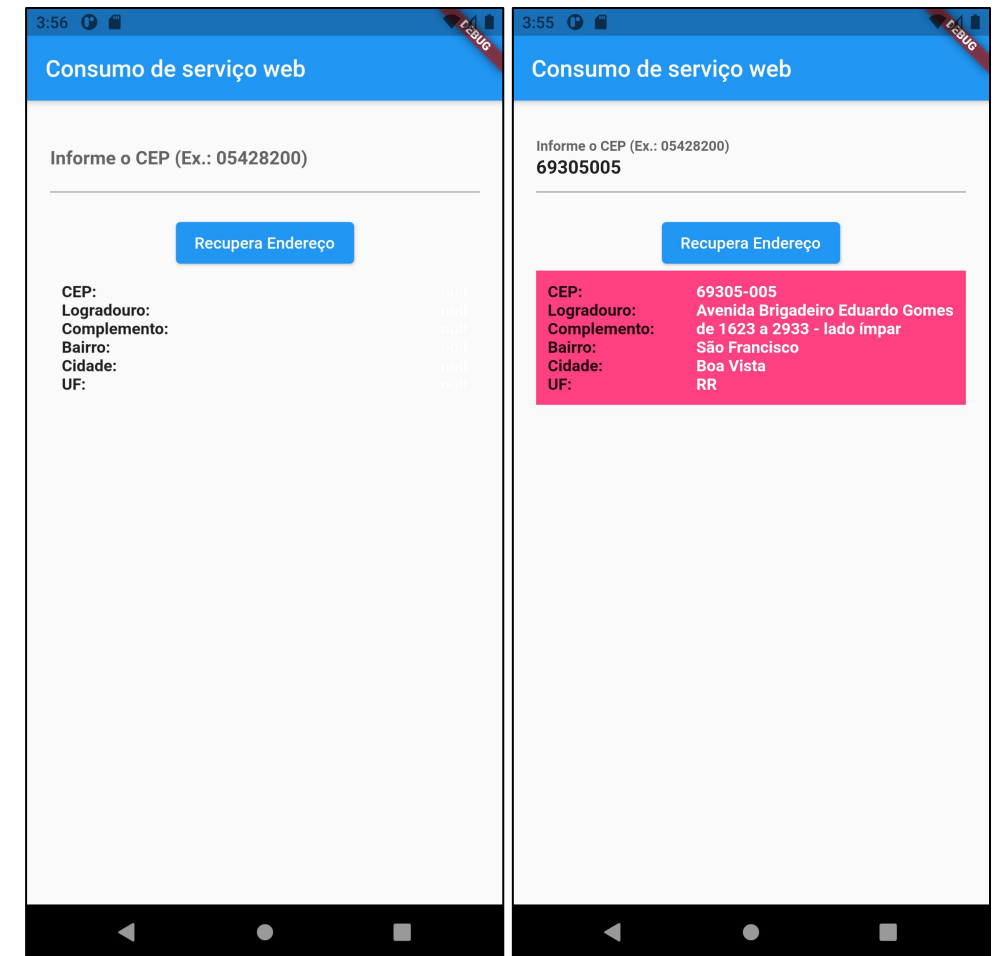
```
import 'package:http/http.dart' as http;

void _recuperarCEP() async{
  String inCEP = _cep.text;
  var url = Uri.parse("https://viacep.com.br/ws/$inCEP/json/");
  http.Response response;

  response = await http.get(url);

  print("Código de Status: " + response.statusCode.toString());
  // print("Resposta: " + response.body);

  setState(() {
    _retorno = json.decode(response.body);
    _corDoFundo = Colors.pinkAccent;
  });
}
```

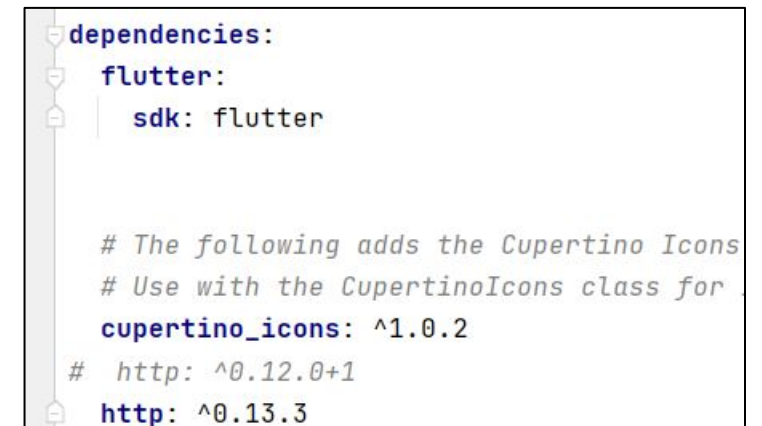


Flutter (Requisições Web)

- Para podermos usar o pacote “http.dart” precisamos configura o arquivo “pubspec.yaml” inserindo a dependência:

`http: ^0.13.3`

- Que indica o pacote e versão dele.
- A configuração é lustrada na imagem ao lado:

A screenshot of a code editor showing the dependencies section of a pubspec.yaml file. The code is as follows:

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  # The following adds the Cupertino Icons  
  # Use with the CupertinoIcons class for  
 /cupertino_icons: ^1.0.2  
  # http: ^0.12.0+1  
  http: ^0.13.3
```

Flutter (Requisições Web)

APP: Preço do Bitcoin

- Link para API:
https://www.blockchain.com/pt/api/exchange_rates_api
- API (URL): <https://blockchain.info/ticker>



The screenshot shows the Blockchain.com website's API section for exchange rates. It includes a search bar, a dropdown for currency (USD), and a section titled 'API de taxas de câmbio'. Below this, it provides information about the API's CORS settings and a sample JSON response for the ticker endpoint.

Blockchain.com

API de Blockchain > Taxas de câmbio

Pesquisar TX, endereço ou bloco USD

API de taxas de câmbio

Preços de mercado e API de taxas de câmbio

Algumas chamadas de API estão disponíveis com **Cabeçalhos CORS** se adicionar um parâmetro `&cors=true` ao pedido GET

URL: <https://blockchain.info/ticker>

Sem parâmetros

Devolve um objeto JSON com os códigos da moeda como chaves. "15m" é o preço de mercado atrasado de 15 minutos, "último" é o preço de mercado mais recente, "símbolo" é o símbolo da moeda.

Resposta: 200 OK, application/json

Utilizamos cookies para melhorar a sua experiência de utilizador. Ao continuar no nosso site, concorda com a nossa [política de privacidade](#).

OK

```
{
  "AUD" : {"15m" : 66474.3, "last" : 66474.3, "buy" : 66474.3, "sell" : 66474.3, "symbol" : "AUD"},
  "BRL" : {"15m" : 257511.03, "last" : 257511.03, "buy" : 257511.03, "sell" : 257511.03, "symbol" : "BRL"},
  "CAD" : {"15m" : 62103.17, "last" : 62103.17, "buy" : 62103.17, "sell" : 62103.17, "symbol" : "CAD"},
  "CHF" : {"15m" : 45184.13, "last" : 45184.13, "buy" : 45184.13, "sell" : 45184.13, "symbol" : "CHF"},
  "CLP" : {"15m" : 3.815769928E7, "last" : 3.815769928E7, "buy" : 3.815769928E7, "sell" : 3.815769928E7, "symbol" : "CLP"},
  "CNY" : {"15m" : 324031.99, "last" : 324031.99, "buy" : 324031.99, "sell" : 324031.99, "symbol" : "CNY"},
  "CZK" : {"15m" : 1049106.72, "last" : 1049106.72, "buy" : 1049106.72, "sell" : 1049106.72, "symbol" : "CZK"},
  "DKK" : {"15m" : 358169.47, "last" : 358169.47, "buy" : 358169.47, "sell" : 358169.47, "symbol" : "DKK"},
  "EUR" : {"15m" : 41564.02, "last" : 41564.02, "buy" : 41564.02, "sell" : 41564.02, "symbol" : "EUR"},
  "GBP" : {"15m" : 35664.73, "last" : 35664.73, "buy" : 35664.73, "sell" : 35664.73, "symbol" : "GBP"},
  "HKD" : {"15m" : 383601.1, "last" : 383601.1, "buy" : 383601.1, "sell" : 383601.1, "symbol" : "HKD"},
  "INR" : {"15m" : 3940649.0, "last" : 3940649.0, "buy" : 3940649.0, "sell" : 3940649.0, "symbol" : "INR"},
  "ISK" : {"15m" : 9855420.09, "last" : 9855420.09, "buy" : 9855420.09, "sell" : 9855420.09, "symbol" : "ISK"},
  "JPY" : {"15m" : 5419455.65, "last" : 5419455.65, "buy" : 5419455.65, "sell" : 5419455.65, "symbol" : "JPY"},
  "KRW" : {"15m" : 5.739740283E7, "last" : 5.739740283E7, "buy" : 5.739740283E7, "sell" : 5.739740283E7, "symbol" : "KRW"},
  "NZD" : {"15m" : 69433.09, "last" : 69433.09, "buy" : 69433.09, "sell" : 69433.09, "symbol" : "NZD"},
  "PLN" : {"15m" : 187034.35, "last" : 187034.35, "buy" : 187034.35, "sell" : 187034.35, "symbol" : "PLN"},
  "RUB" : {"15m" : 3585064.89, "last" : 3585064.89, "buy" : 3585064.89, "sell" : 3585064.89, "symbol" : "RUB"},
  "SEK" : {"15m" : 455267.46, "last" : 455267.46, "buy" : 455267.46, "sell" : 455267.46, "symbol" : "SEK"},
  "SGD" : {"15m" : 65893.95, "last" : 65893.95, "buy" : 65893.95, "sell" : 65893.95, "symbol" : "SGD"},
  "THB" : {"15m" : 1598276.23, "last" : 1598276.23, "buy" : 1598276.23, "sell" : 1598276.23, "symbol" : "THB"},
  "TWD" : {"15m" : 1787872.06, "last" : 1787872.06, "buy" : 1787872.06, "sell" : 1787872.06, "symbol" : "TWD"},
  "USD" : {"15m" : 49359.98, "last" : 49359.98, "buy" : 49359.98, "sell" : 49359.98, "symbol" : "USD"}
}
```

Flutter (Requisições Web)

APP: Preço do Bitcoin

```
void _cotacaoBitcoin() async {  
  var url = Uri.parse("https://blockchain.info/ticker");  
  http.Response response;  
  
  response = await http.get(url);  
  _retorno = json.decode(response.body);  
  
  print("Código de Status: " +  
response.statusCode.toString());  
  print("Cotação: ${_retorno['BRL']['buy']}");  
  
  setState(() {  
    _cotacao = "R\$ ${_retorno['BRL']['buy']}";  
  });  
}
```



Flutter (Exercícios)

1. Adicionando funcionalidade na App ATM Consultoria:

- Adicione um Radio button com as opções de Real, Euro e Dólar;
 - O botão Radio estará configurado como padrão no Real, no entanto, o usuário poderá mudar a cotação para Euro ou dólar.
 - Quando a requisição for feita a cotação a ser exibida deve esta de acordo com a opção que o usuário selecionou.

