

INSTITUTO MAUÁ DE TECNOLOGIA



# Linguagens 2

Introdução ao Back-end

Prof. Tiago Sanches da Silva

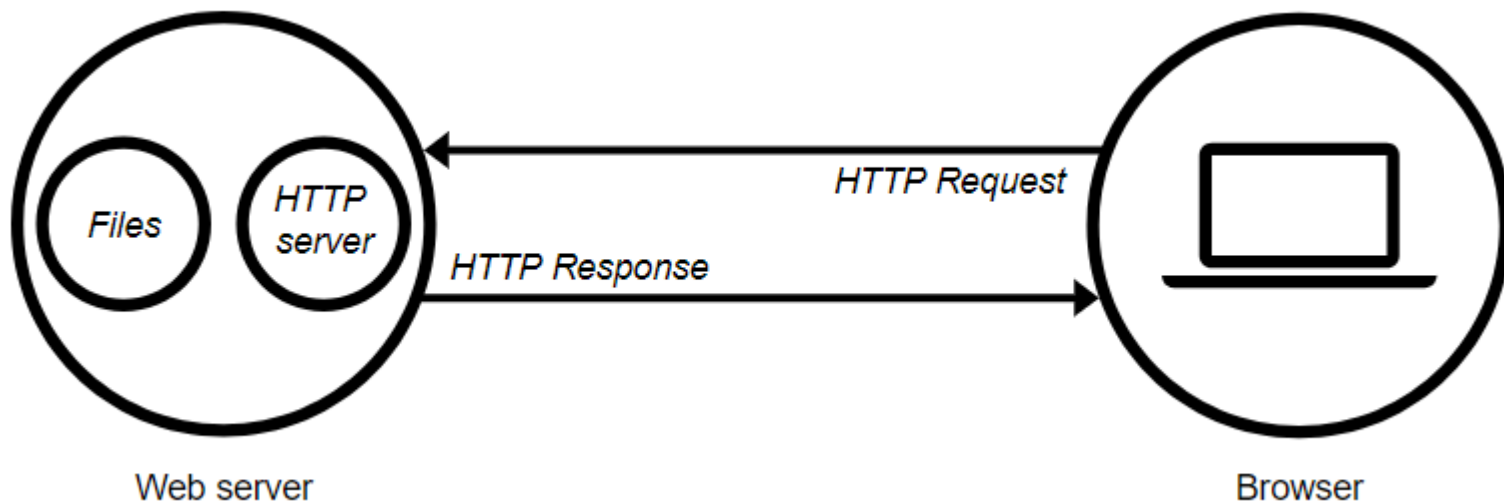
Web server

"Servidor web (web server)" pode referir ao **hardware** ou ao **software**, ou ambos trabalhando juntos.

Em um nível mais básico, o navegador fará uma requisição utilizando o **protocolo HTTP** sempre que necessitar de um arquivo hospedado em um servidor web. Quando a requisição alcançar o servidor web correto (hardware), o servidor HTTP (software) enviará o documento requerido, também via HTTP.

# Web server

“Quando a requisição alcançar o servidor web correto (hardware), o **servidor HTTP** (software) enviará o documento requerido, também via HTTP.”



Um servidor HTTP precisa “saber” interpretar requisições HTTP, pois toda “conversa” entre cliente e servidor é realizada através deste protocolo.

HTTP

# HTTP (Hypertext Transfer Protocol)

Como o próprio nome indica, o HTTP especifica como transferir documentos de hipertexto entre dois computadores.

**HTTP é um protocolo textual sem estado.**

```
GET /ExemploWebServer.html HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/msword, /* application/*
Accept-Language: en-gb,pt;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4324)
Host: localhost:79
Connection: Keep-Alive
Cookie: infoview_userCultureKey=useBrowserLocale
```

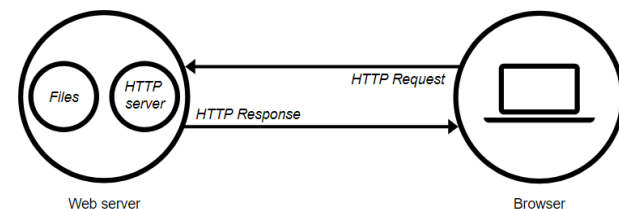
# HTTP (Hypertext Transfer Protocol)

## Requisição

```
GET /ExemploWebServer.html HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-gb,pt;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
Host: localhost:79
Connection: Keep-Alive
Cookie: infoview_userCultureKey=useBrowserLocale
```

## Resposta

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.1
X-Powered-By: ASP.NET
Date: Thu, 11 Aug 2018 01:00:51 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Thu, 12 Aug 2017 13:02:0 GMT
ETag: "cd3d2380c61:br9"
Content-Length: 54
<html>
<body>
  <p> Apenas um Exemplo </p>
</body>
</html>
```



protocolo textual

# Protocolo?

Um protocolo é um conjunto de regras para comunicação entre dois computadores.

**Vamos combinar ações dado alguns campos chave/valor que são passados em texto plano!**

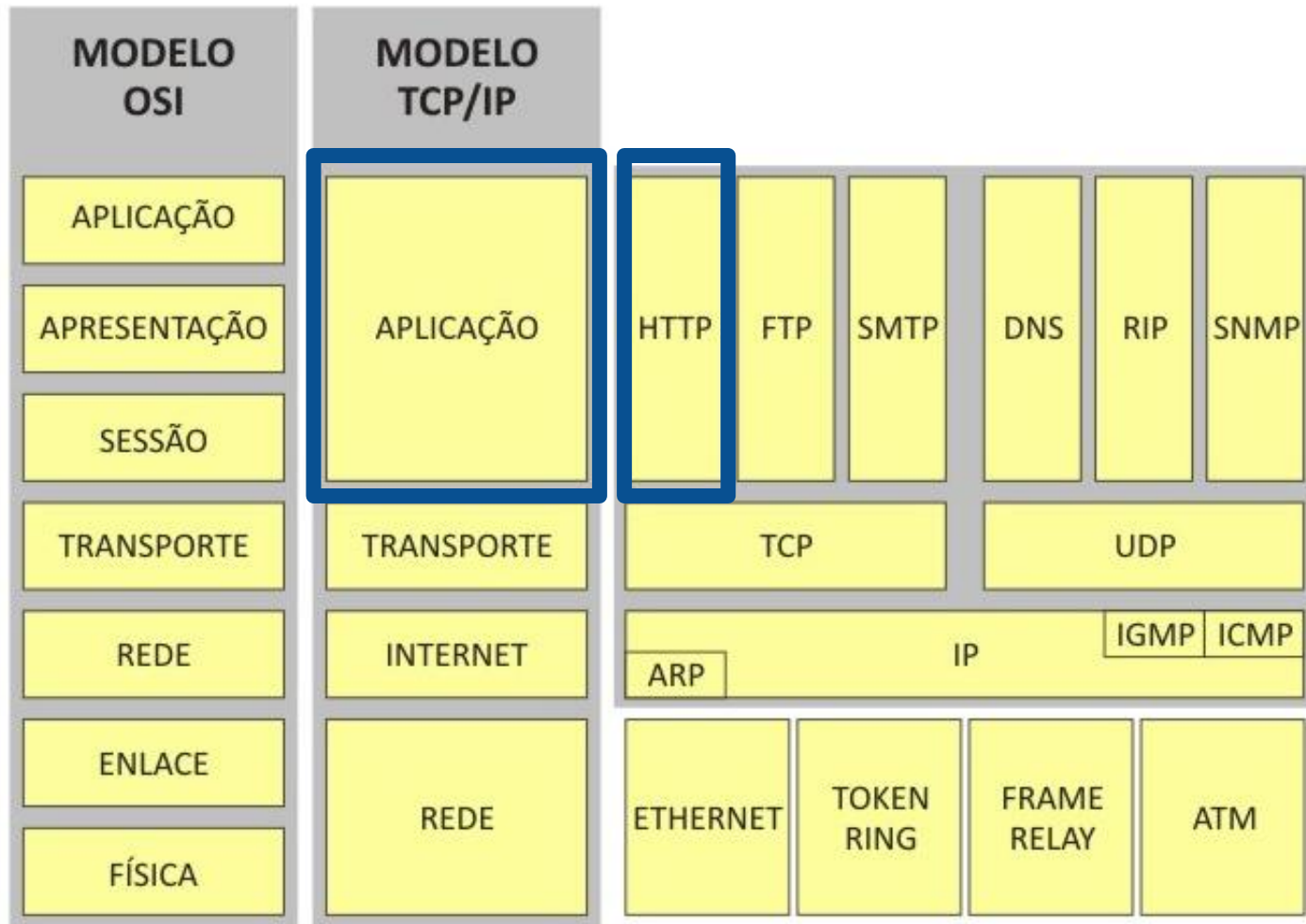
RFC 7234

<https://tools.ietf.org/html/rfc7234#page-12>





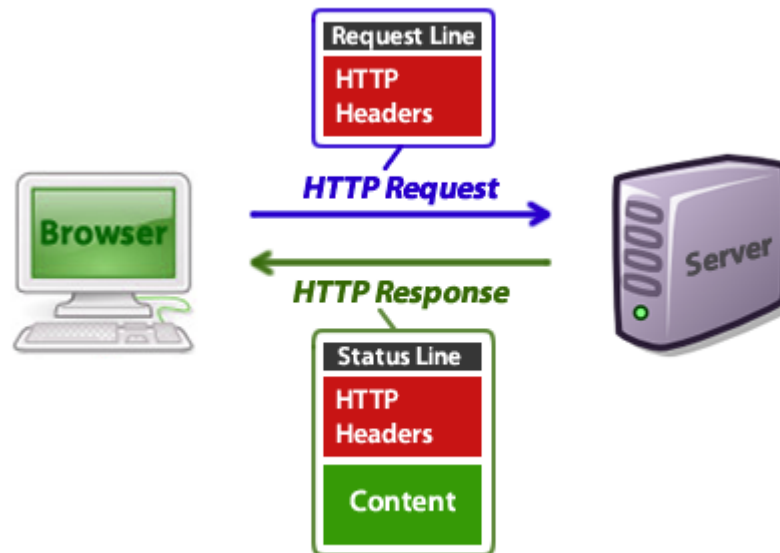
# Protocolo HTTP



# Servidor HTTP

# Servidor HTTP

Um servidor HTTP então é um software que é capaz de entender e processar requisições HTTP, bem como responde-las utilizando o protocolo (regras) HTTP definido no RFC 7234.



# Discussão em sala

Podemos criar nosso servidor HTTP através de sockets utilizando C, Java, Pascal ou qualquer outra linguagem?

Precisamos?

The logo for NGINX, featuring the word "NGINX" in a green, stylized, blocky font.

# Servidor HTTP

Onde encontrar? É Gratuito?

- Apache: <http://httpd.apache.org/>
- NGINX: <https://www.nginx.com/>

The NGINX logo is displayed in a bold, green, sans-serif font. The letters are slightly stylized, with the 'i' and 'n' having unique shapes.The word "Apache" is written in a bold, black, sans-serif font, positioned directly below the feather logo.

Web server estático ou dinâmico?

Qual a diferença entre web server (servidor web) estático e o dinâmico?

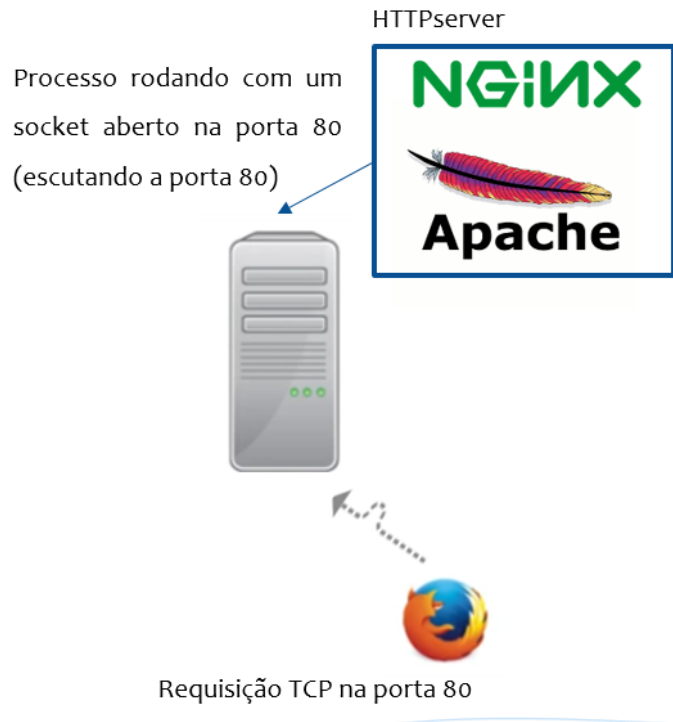
Um servidor estático significa que ele entrega o conteúdo pedido “da forma que está” (“As is”). E dinâmico significa que o servidor processa o conteúdo ou o gera a partir de um banco de dados.

A grosso modo, executa algum script ou processamento no servidor?

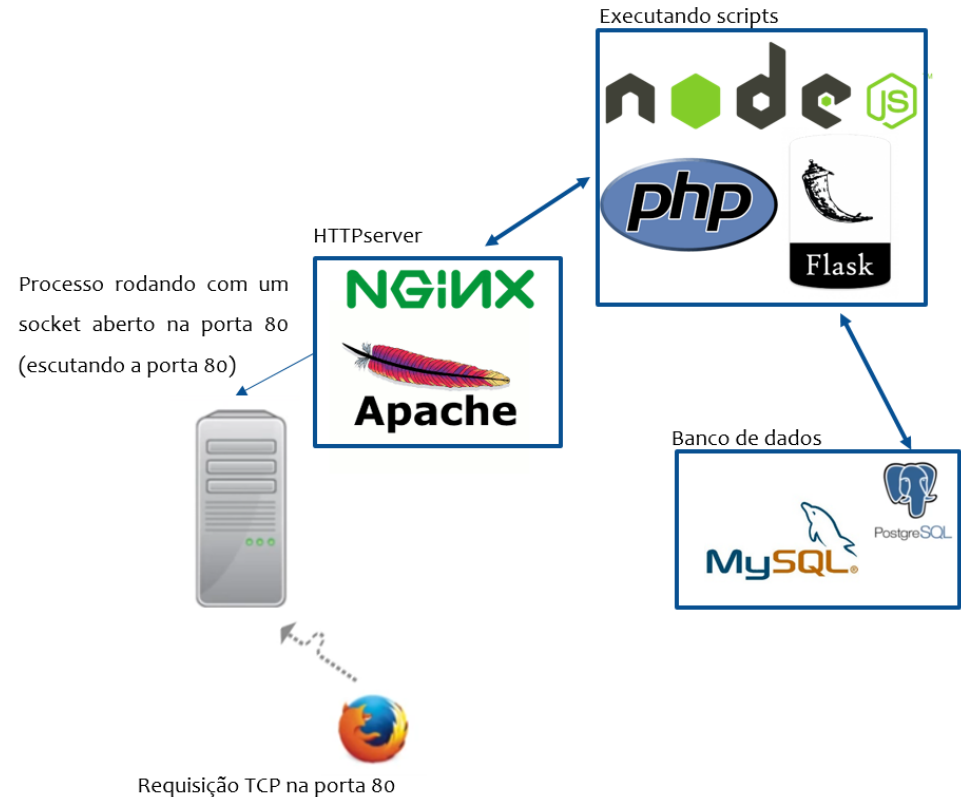
- Sim: **dinâmico**.
- Não, apenas responde com a pagina html que esta no servidor: **estático**.

# Web server – topologia

## Estático



## Dinâmico



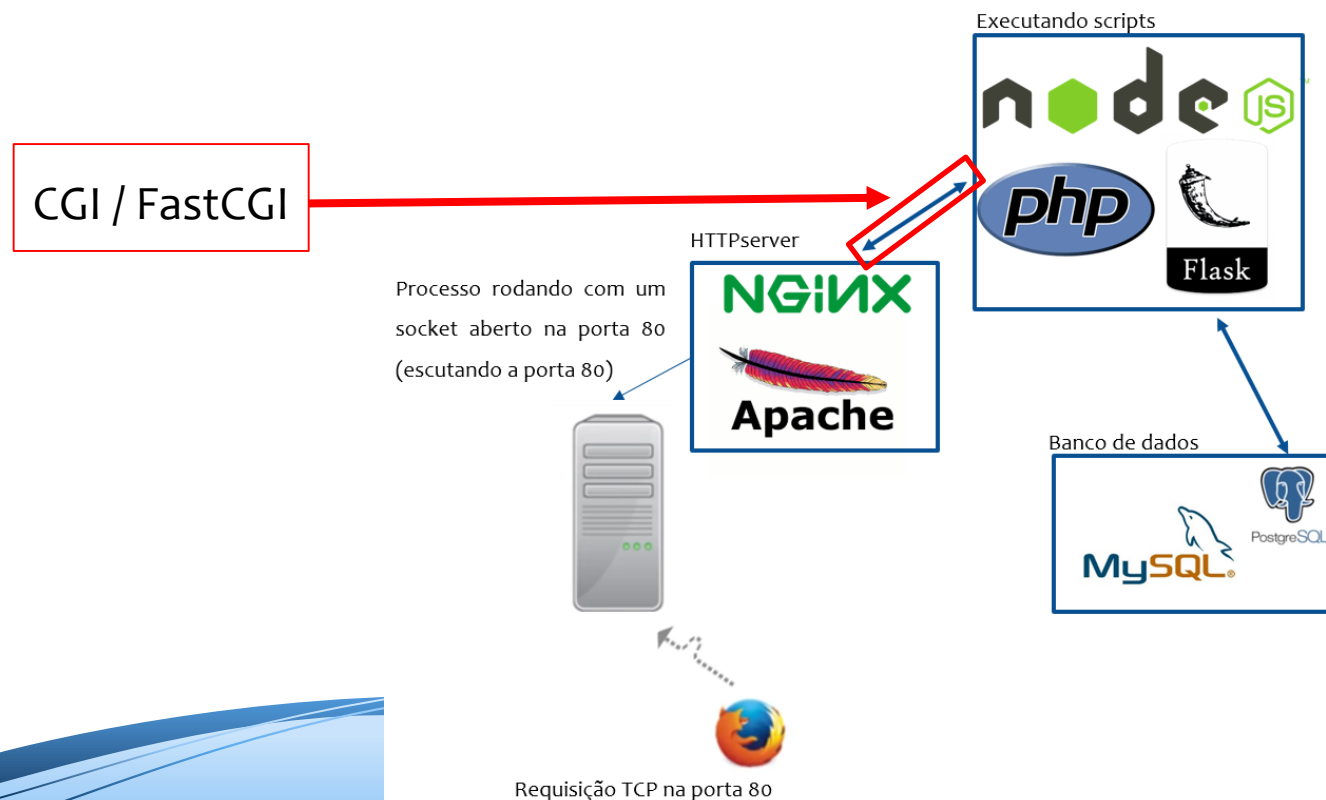


# CGI

(Common Gateway Interface)

# CGI - Common Gateway Interface

Common Gateway Interface (CGI) oferece um protocolo padrão para servidores web executar programas que “rodam” como aplicativos de console (também chamados de programas de interface de linha de comando) em execução em um servidor que gera respostas dinamicamente.



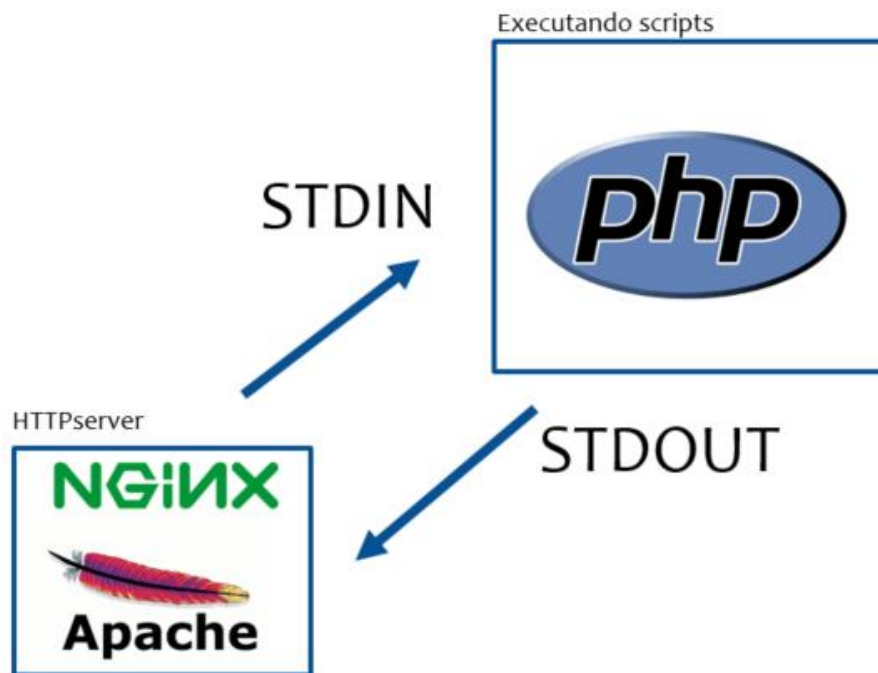
# CGI - Common Gateway Interface

De um modo bem simples, podemos dizer que o Common Gateway Interface é um “acordo” entre os servidores HTTP e as aplicações Web. Por baixo dos panos, o servidor Web vai informar uma série de parâmetros para o seu programa, e é dever do seu programa entregar uma resposta “bem formada” para o servidor Web.

Isso quer dizer que, para o **CGI**, não importa qual linguagem ou banco de dados o seu programa está usando. Para ele, importa a **passagem dos parâmetros (STDIN)** e a **resposta (STDOUT)**.

# CGI - Common Gateway Interface

Isso quer dizer que, para o **CGI**, não importa qual linguagem ou banco de dados o seu programa está usando. Para ele, importa a **passagem dos parâmetros (STDIN)** e a **resposta (STDOUT)**.



# FastCGI

O FastCGI é uma variação da anterior Common Gateway Interface (CGI); O principal objetivo da FastCGI é reduzir a sobrecarga associada à interface do servidor HTTP e dos programas CGI, permitindo que um servidor manipule mais solicitações de páginas da Web de uma só vez.

Leitura complementar - excelente artigo sobre CGI x FastCGI:

**<https://klauslaube.com.br/2012/11/02/entendendo-o-cgi-fastcgi-e-wsgi.html>**



# Difícil configurar?

Vamos olhar um exemplo de configuração do PHP utilizando o FastCGI no NGINX:

<https://www.nginx.com/resources/wiki/start/topics/examples/phpfcgi/>

```
location ~ [^/]\.php(/|$) {
    fastcgi_split_path_info ^(.+?\.php)(/.*)$;
    if (!-f $document_root$fastcgi_script_name) {
        return 404;
    }

    # Mitigate https://httpoxy.org/ vulnerabilities
    fastcgi_param HTTP_PROXY "";

    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;

    # include the fastcgi_param setting
    include fastcgi_params;

    # SCRIPT_FILENAME parameter is used for PHP FPM determining
    # the script name. If it is not set in fastcgi_params file,
    # i.e. /etc/nginx/fastcgi_params or in the parent contexts,
    # please comment off following line:
    # fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}
```

O que iremos utilizar?



# O que iremos utilizar?

INICIALMENTE iremos utilizar um servidor HTTP Apache configurado com PHP e para banco de dado o MySQL, essa é a configuração mais amigável para entrar no “mundo do back-end”.



# O que iremos utilizar?

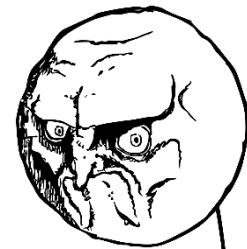
Vamos ter que configurar tudo do zero?  
Não!



PHP

O PHP ("**P**HP: **H**ypertext **P**reprocessor", originalmente Personal Home Page) é uma das linguagens de programação mais populares para o desenvolvimento web. Ela se destaca por ser uma linguagem simples para quem é iniciante e, ao mesmo tempo, por oferecer muitos recursos a programadores de longa data.

O PHP tem, ainda, a capacidade de se misturar com a linguagem HTML e, dessa forma, transformar qualquer site estático em dinâmico.



O PHP é uma linguagem server-side, ou seja, tudo é processado no servidor.

<?php    ?>


## Links interessantes


- Documentação oficial: <http://php.net/manual/en/index.php>
- Tutoriais:
  - <https://www.tutorialspoint.com/php7/>
  - <https://www.tutorialrepublic.com/php-tutorial/>
- Cheat sheet:
  - <http://overapi.com/php> (ÓTIMO)
  - <https://websitesetup.org/php-cheat-sheet/>


Onde queremos chegar?  
Web Services

# Exemplo

<https://www.skyscanner.com.br/>

 Passagens aéreas

 Hotéis

 Aluguel de carros

☒ Ida e volta ☐ Só ida ☐ Várias cidades

De 

São Paulo - Congonhas (CGH)

☐ Adicionar aeroportos próximos

☐ Apenas voos sem conexão

Para 

*País, cidade ou aeroporto*

☐ Adicionar aeroportos próximos

Ida 

03/03/2017

Volta 

04/03/2017

Classe de cabine e Passageiros 

1 adulto, Econômica

Mapa

Buscar →

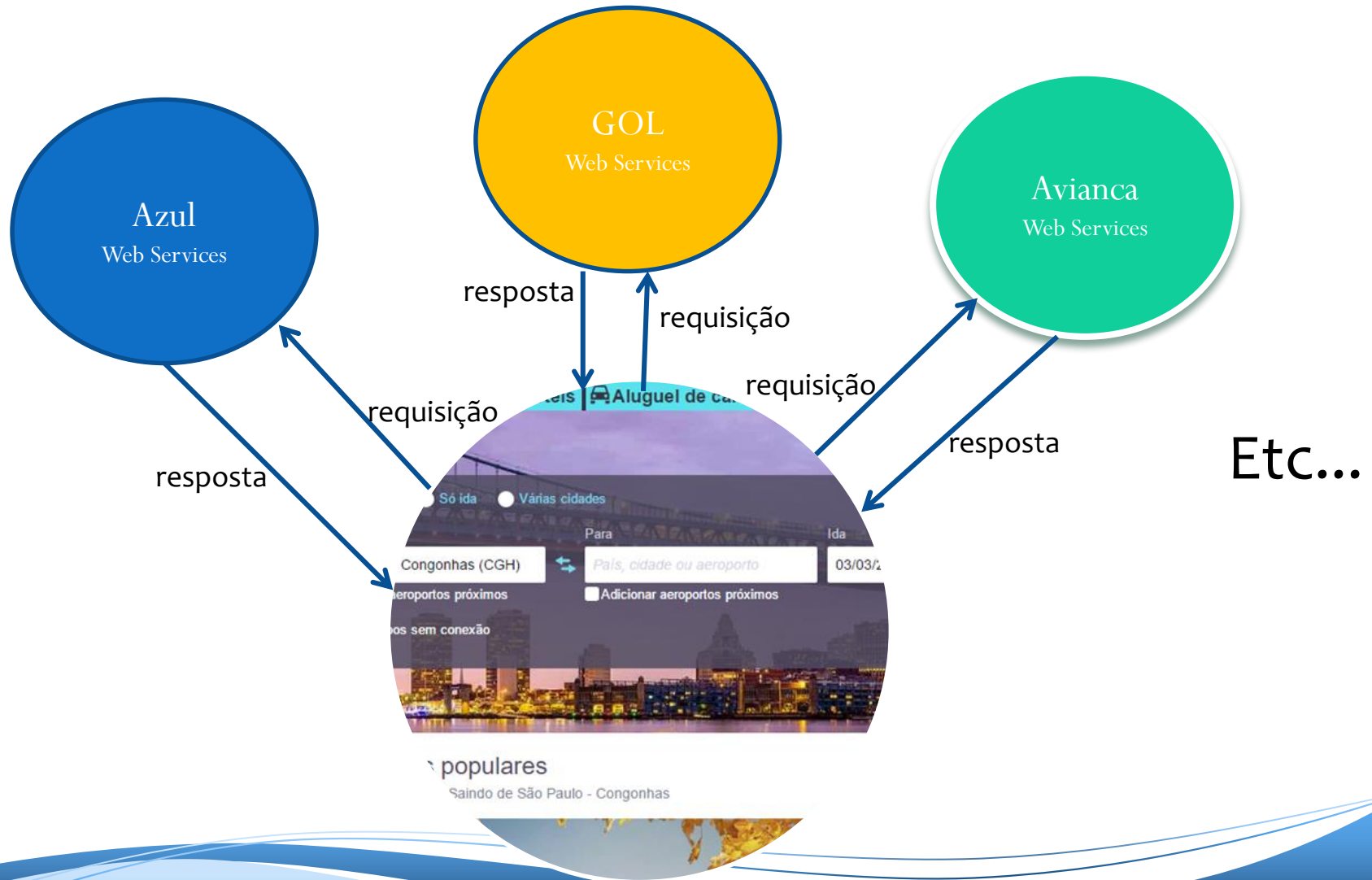
## Destinos populares

junho de 2017 Saindo de São Paulo - Congonhas



# Exemplo

<https://www.skyscanner.com.br/>





# Exemplo

<https://www.skyscanner.com.br/>

529 de 1148 resultados (mostrar tudo)

Ordenar

Preço por adulto

123...53>

Avianca

07:45

CGH

1h 05m

Direto

08:50

GIG

Avianca

09:30

GIG

1h 10m

Direto

10:40

CGH

Avianca

18:00

CGH

1h 10m

Direto

19:10

GIG

Avianca

09:30

GIG

1h 10m

Direto

10:40

CGH

Azul

06:10

CGH

3h 35m

1 conexão CNF

09:45

SDU

Azul

07:15

SDU

6h

1 conexão CNF

13:15

CGH

Azul

10:10

CGH

2h 50m

1 conexão CNF

13:00

SDU

Azul

07:15

SDU

6h

1 conexão CNF

13:15

CGH

4 ofertas a partir de

R\$ 632

Veja mais →

4 ofertas a partir de

R\$ 712

Veja mais →

3 ofertas a partir de

R\$ 713

Veja mais →

3 ofertas a partir de

R\$ 713

Veja mais →


# Comunicação com as Web Services

# Comunicação com as Web Services

É necessário uma padronização que garanta **a comunicação** entre o Web service e o sistema que realizou a requisição.

Essa comunicação, por sua vez, segue alguns protocolos, principalmente relacionados **ao formato da transmissão de dados**, o que permite que, uma vez que um sistema implemente essas regras, qualquer outro sistema que siga o mesmo protocolo seja capaz de se comunicar com ele.

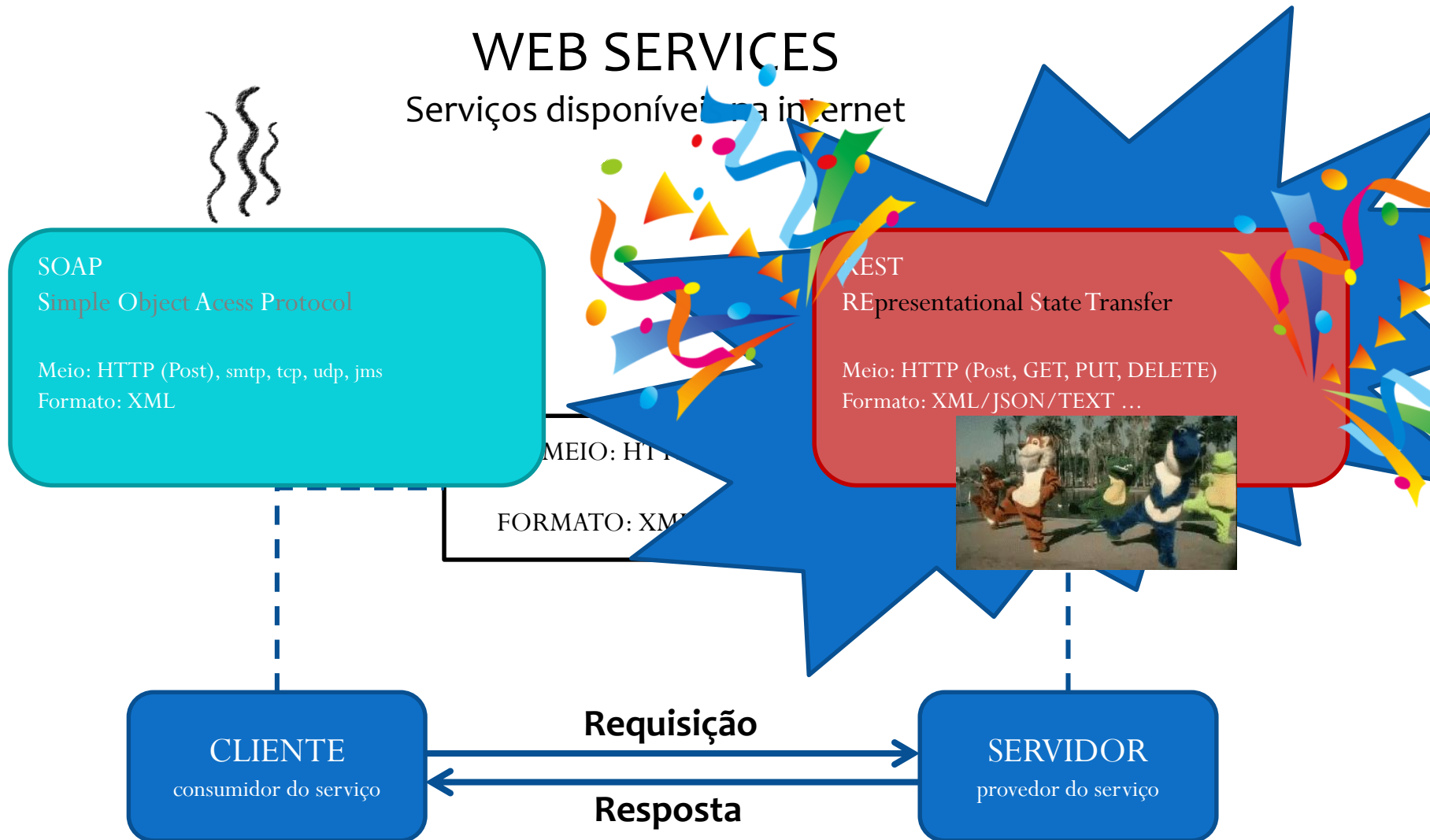
Entre essas padronizações estipuladas, as duas de maior destaque são o protocolo **SOAP** e o modelo de design **REST**, as quais iremos discutir nos próximos tópicos.



# Tipos de Web Service

## WEB SERVICES

Serviços disponíveis na internet



Json

JSON ou JavaScript Object Notation é um padrão aberto baseado em texto projetado para intercâmbio de dados legível. **Baseado em chave : valor.**

O formato JSON foi originalmente especificado por Douglas Crockford, e é descrito no RFC 4627.

- JSON é fácil de ler e escrever.
- É um formato de intercâmbio baseado em texto leve.
- JSON é independente da linguagem.

# Json

Os campos são separados por virgula e é possível criar vetores e formas mais complexas com o encadeamento de Jsons.

```
{
  "id": 1,
  "name": "A green door",
  "price": 12.50,
  "tags": ["home", "green"]
}
```

```
{
  "book": [
    {
      "id": "01",
      "language": "Java",
      "edition": "third",
      "author": "Herbert Schildt",
    },
    {
      "id": "07",
      "language": "C++",
      "edition": "second",
      "author": "E.Balagurusamy",
    }
  ]
}
```

Exemplos mais complexos podem ser encontrados: <http://json.org/example.html>

Verificador Json (Projeto Lint): <http://jsonlint.com/>

API



API significa Application Programming Interface (Interface de Programação de Aplicativos) e nada mais é do que uma interface simplificada e padronizada que é desenvolvida com o intuito de tornar a sua aplicação mais comunicável.

O foco principal das APIs é fornecer a outros programadores um método simples para que eles possam construir aplicações que se comuniquem com a sua.

Geralmente elas são escritas em um padrão aberto e com uma documentação que é disponibilizada livremente. Hoje temos diversos serviços online que liberam suas APIs para que outros softwares possam fazer uma integração, como é o caso de Amazon, Flickr, YouTube e Twitter, por exemplo. Quando uma API utiliza comunicação via web, nós a chamamos de [Web Service](#).

# API

Onde posso encontrar algumas APIs?

Facebook:

<https://developers.facebook.com/docs/>

<https://developers.facebook.com/docs/graph-api/using-graph-api>

Google maps:

<https://developers.google.com/maps/>

<https://developers.google.com/maps/web-services/client-library>

Twitter:

<https://dev.twitter.com/rest/public>

OKOK... Vc quer dominar o mundo?

<https://www.programmableweb.com/category/all/apis>

<https://apigee.com/console/instagram>

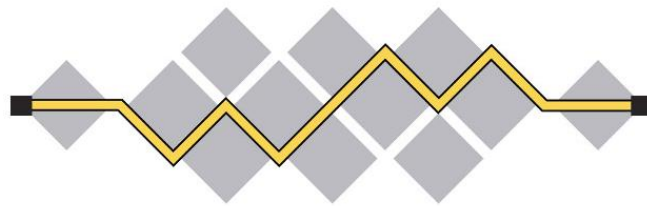
Relaxem, voltaremos aqui com mais calma.

RFC

Acrônimo de **Request for Comments** (ou "pedido para comentários" em português), as **RFCs** são documentos técnicos desenvolvidos e mantidos pelo **IETF** (**Internet Engineering Task Force**), instituição que especifica os padrões que serão implementados e utilizados em toda a internet.

Existe **RFC** para tudo na internet, e cada um deles deve detalhar o funcionamento de todos os aspectos do protocolo proposto.

Pra tudo mesmo? TCP? SMTP? STREAMING? E-MAIL? HTTP? IPSEC? AES?



Perguntas?