



Programador Web



PHP - Conceitos Básicos da WEB

Adriel Sales

Conceitos básicos da WEB

INTERNET

- Conjunto de redes interligadas que liga milhares de redes independentes de computadores, utilizando um conjunto de tecnologias e protocolos comuns.

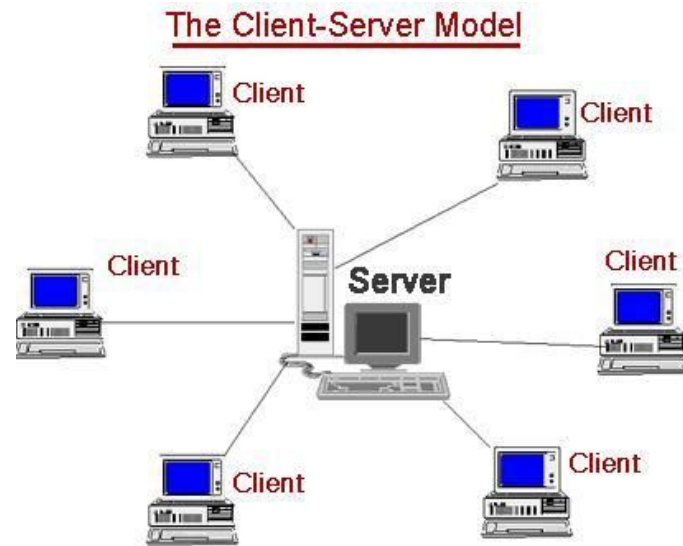
TCP/IP

- Protocolo de comunicação standard, que permite a comunicação entre sistemas de computadores, independentemente dos sistemas operacionais, do hardware e dos próprios canais de comunicação (linha telefónica, fibra óptica, linha dedicadas).
- O IP funciona assim como o identificador de um dispositivo ligado à internet. **Exemplo: 195.192.2.21**

Arquitetura WEB - Cliente-Servidor

Do lado do cliente, os usuários utilizam um programa chamado de navegador (em inglês, browser) para manipular as páginas HTML.

No lado do servidor, fica a camada de armazenamento dos dados, onde são executados um ou mais serviços ou programas que compartilham recursos com os clientes.



Neste modelo arquitetural, um cliente nunca se comunica diretamente com outro cliente, pois toda a comunicação ocorre diretamente com o servidor.

Conceitos básicos da WEB - DNS / WWW

DNS

- **DNS (Domain Name System)** - Sistema de gestão de nomes com vista a traduzir domínios em endereços IP. Este sistema foi criado para facilitar a memorização de endereços na internet.
- Exemplo: **g1.com.br** <===> **186.192.90.5**

WWW

- A World Wide Web (ou Web) é um sistema de documentos e recursos identificados através de **URLs** que se tornou no serviço mais popular da internet.

Conceitos básicos da WEB - HTTP



Um protocolo é um conjunto de regras que descrevem a forma como é efetuada a transmissão e recepção de dados (COMUNICAÇÃO), nomeadamente em uma rede de computadores.

HTTP (HyperText Transfer Protocol)

- É o protocolo mais utilizado na Web e foi criado por Tim Berners-Lee, inventor do hipertexto, para ligar documentos entre si através de hiperlinks.

HTTPS (HyperText Transfer Protocol over Secure Socket)

- Implementação do protocolo HTTP sobre uma camada de **segurança SSL** que permite a troca de dados mais segura através da encriptação da informação.

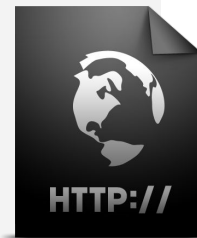
Conceitos básicos da WEB - Comunicação HTTP

HTTP - REQUEST

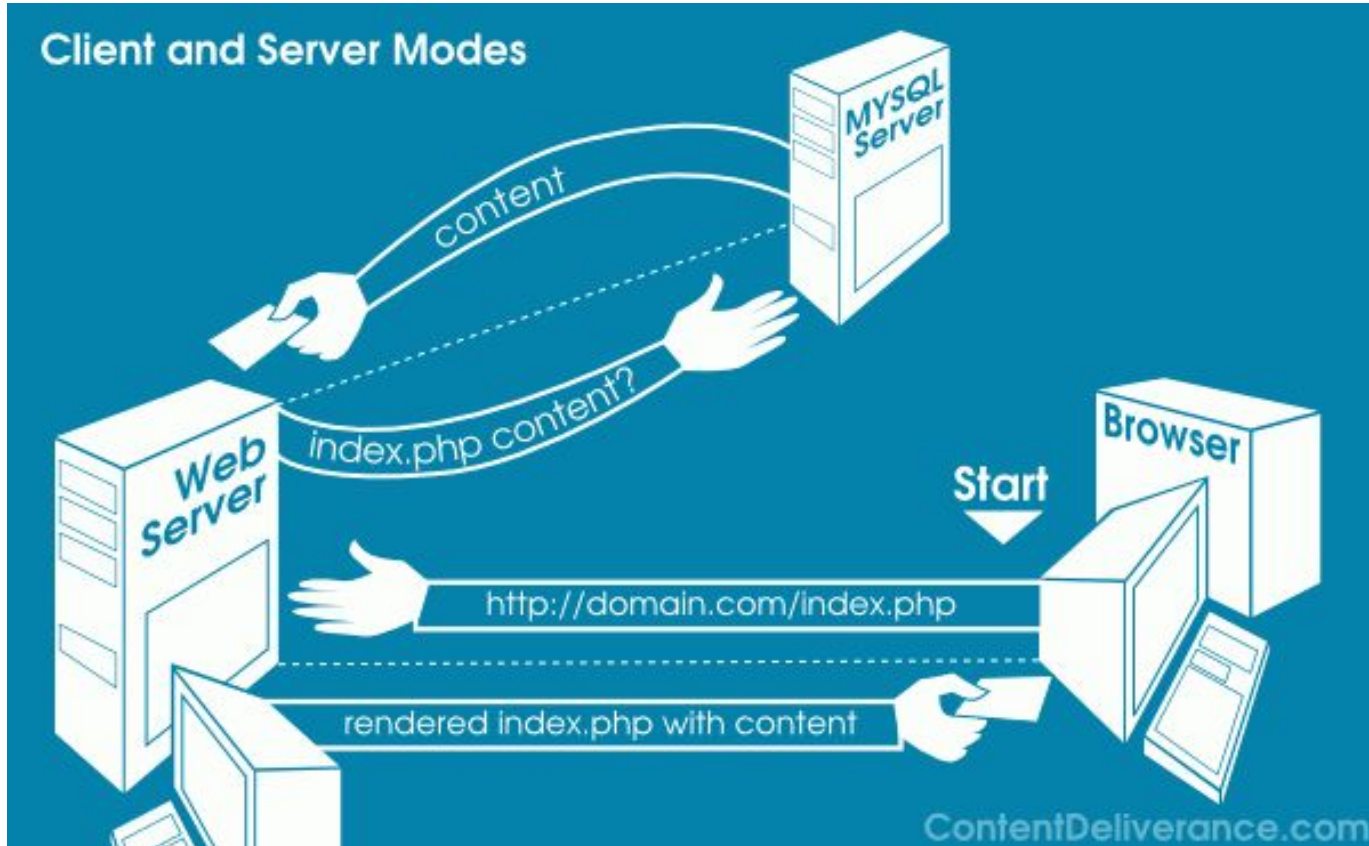
- Ao enviar uma mensagem solicitando um **recurso**, o navegador (cliente) aponta sua requisição para um determinado endereço na rede – o qual deve ser único – **chamado URL**. Este recurso pode ser um arquivo, uma impressora ou até mesmo outro aplicativo.

HTTP - RESPONSE

- Ao receber uma requisição, o servidor, imediatamente, a interpreta, **prepara uma mensagem** e envia **uma resposta ao cliente**.



HTTP Request / Response



Conceitos básicos da WEB - URL



URL

- Um **URL (Uniform Resource Locator)** é um endereço composto por um conjunto de caracteres alfanuméricos que identifica um documento na web.
- Exemplo:
<http://www.pb.senac.br/images/banner.png>

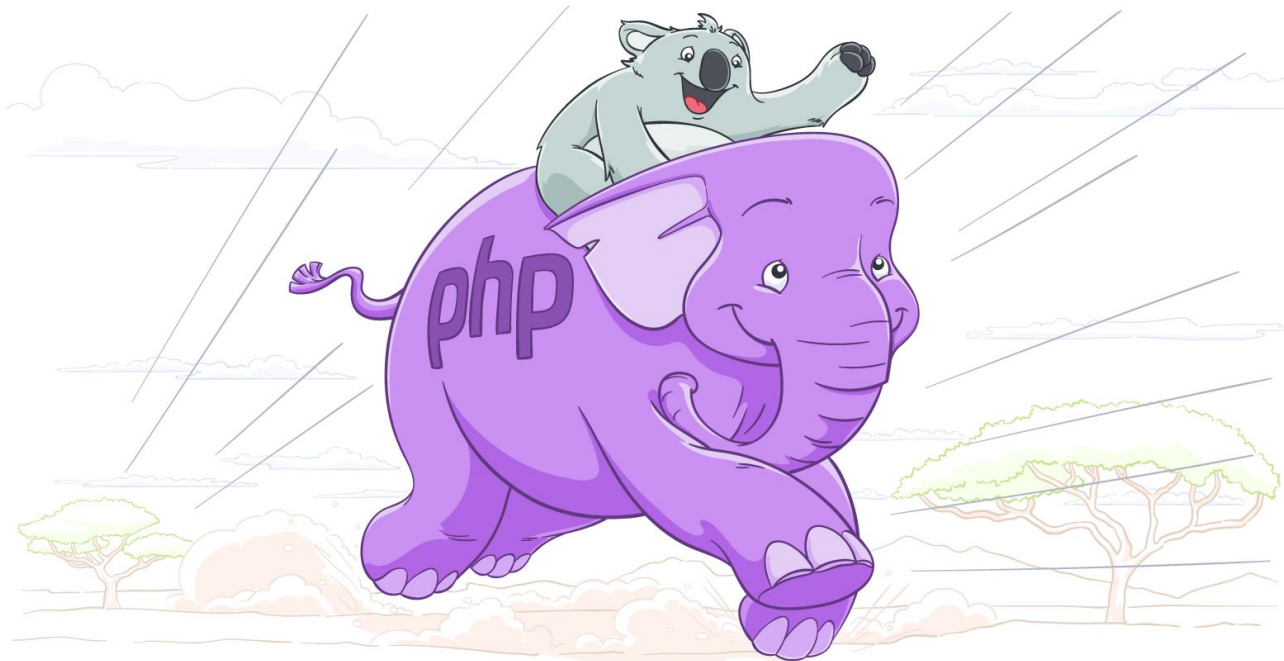
Conceitos básicos da WEB - URL



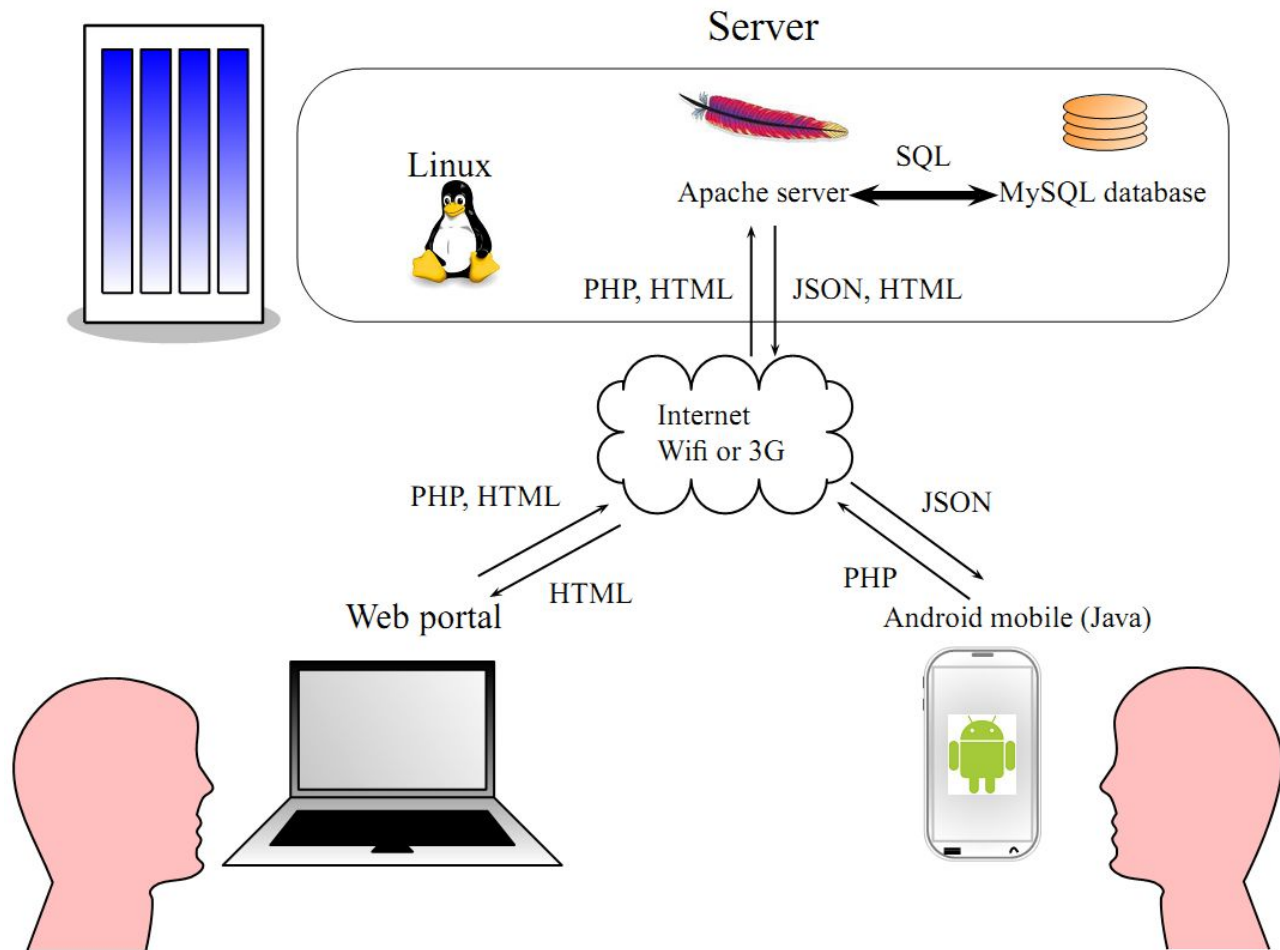
1. **Protocolo** utilizado na requisição para realizar a conexão.
2. **Nome do servidor** ao qual se pretende recuperar o recurso desejado.
3. **Pasta/diretório** onde está localizado o arquivo (sujeito à níveis).
4. **Nome do arquivo** ou página que se quer recuperar.

PHP no mundo Web

Entenderemos
passo a passo o
fluxo de um **HTTP
request /
response**,
utilizando **PHP**
como **linguagem
de programação**.

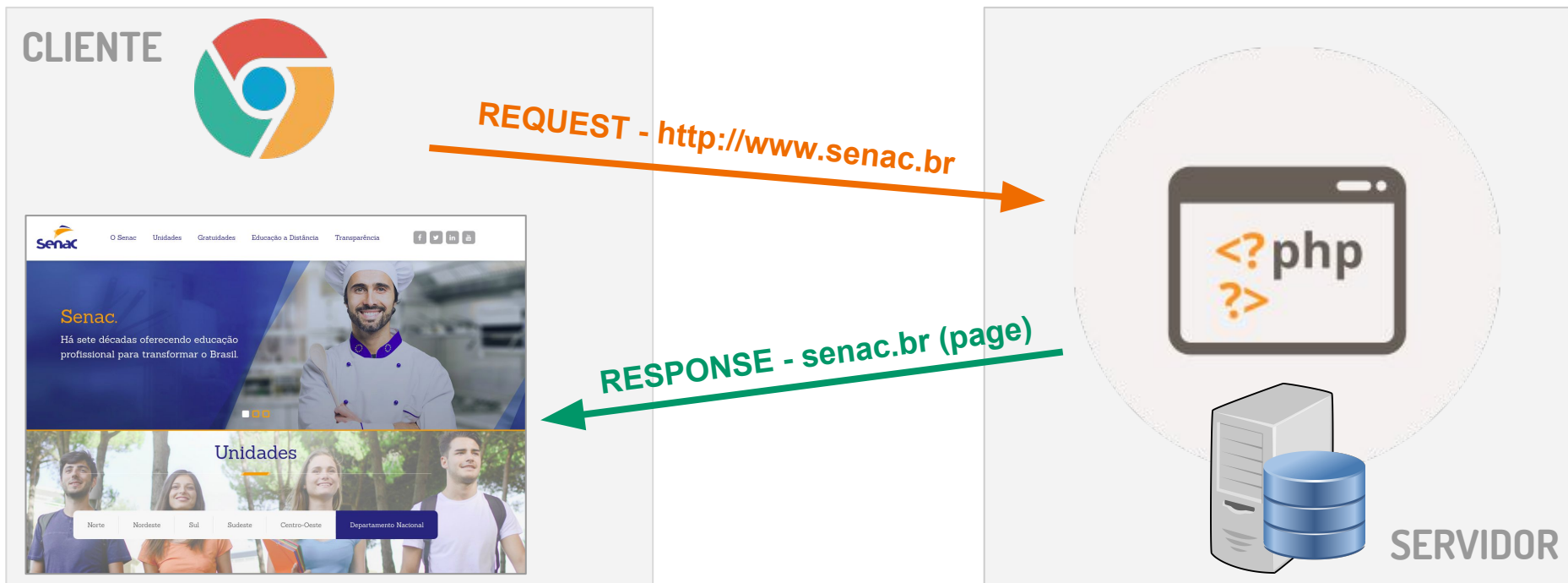


Fluxo de um HTTP request / response.



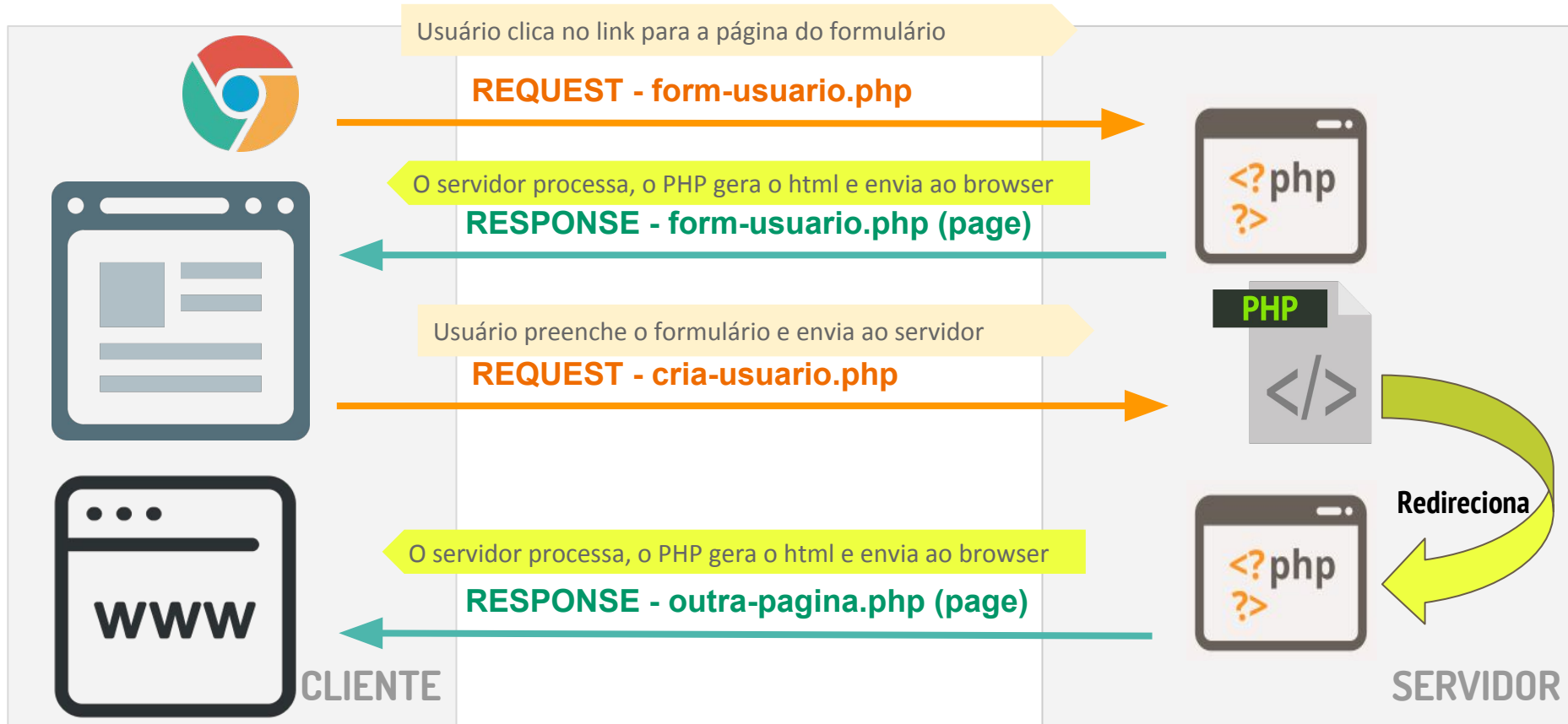
REQUEST/RESPONSE para uma página web

O cliente faz uma requisição por um recurso, a partir de uma URL. O PHP, no lado do servidor interpreta, escreve a mensagem e responde, entregando uma página web.



Requisições a partir de um formulário

(para outra página)



Requisições a partir de um formulário

(para outra página)



Loja Programador Web

Cadastrar Usuário

Nome

Descrição

Salvar

CLIENTE

Usuário clica no link para a página do formulário

REQUEST - form-usuario.php



```
<form action="form-usuario.php" method="get">
  <label for="nome">Nome</label>
  <input type="text" name="nome">
  <label for="descricao">Descrição</label>
  <textarea name="descricao"></textarea>
  <input type="submit" name="salvar" value="Salvar">
</form>
```

O servidor processa, o PHP gera o html e envia ao browser

RESPONSE - form-usuario.php (page)

SERVIDOR

Requisições a partir de um formulário

(Solicitando Página com o formulário)

Usuário preenche o formulário e envia ao servidor

REQUEST - criar-usuario.php

PHP



Loja Programador Web

Cadastrar Usuário

Nome

João Maria

Descrição

Administrador

Salvar



CLIENTE

O PHP possui variáveis globais, em forma de um **array associativo**, que conseguem capturar os dados enviados pelo formulário, considerando **a chave e o valor** de cada campo no form. São elas:

\$_GET

\$_POST

\$_FILES (para arquivos)

SERVIDOR

Requisições a partir de um formulário

Chaves

```
<form action="form-usuario.php" method="get">
  <label for="nome">Nome</label>
  <input type="text" name="nome">
  <label for="descricao">Descrição</label>
  <textarea name="descricao"></textarea>
  <input type="submit" name="salvar" value="Salvar">
</form>
```

As Chaves são os nomes dos inputs, textarea, select, etc, no formulário.

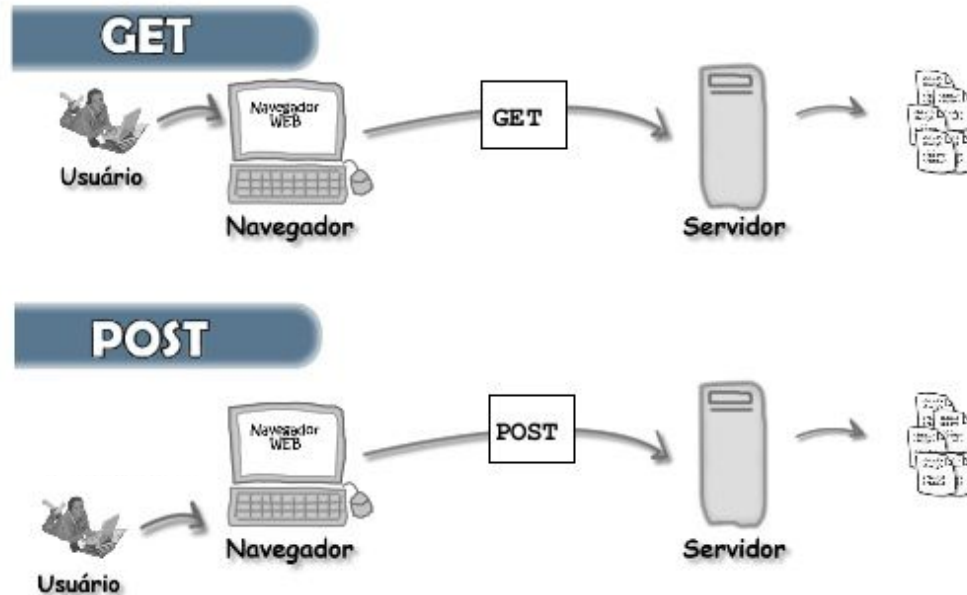
The screenshot shows a web form titled "Loja Programador Web". It has a light blue header bar with the text "Cadastrar Usuário". Below this, there are two input fields. The first is labeled "Nome" and contains the text "João Maria". The second is labeled "Descrição" and contains the text "Administrador". Below these fields is a green button labeled "Salvar". Two arrows originate from the text "Os Valores são os dados informados pelo Usuário." at the bottom and point to the "João Maria" and "Administrador" text in the respective input fields.

Os Valores são os dados informados pelo Usuário.

Ao enviar, o php guardaria os dados no array associativo \$_GET, automaticamente:

```
$_GET["nome"]="João Maria", "descricao"]="Administrador"];
```


Métodos HTTP: usar **GET** ou **POST**?



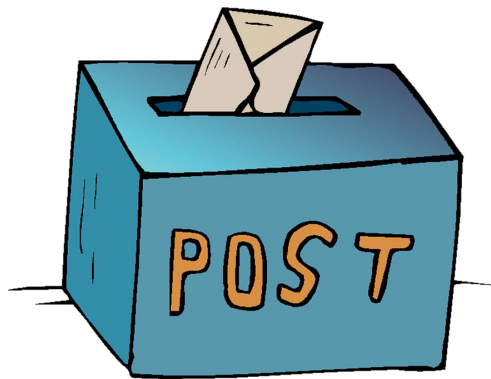
Como, quando, onde?...

Métodos HTTP: usar **GET** ou **POST**?

- Capacidade limitada a 1024 caracteres;
- Passa informações para outra página através da URL.
 - `http://www.meusite.com.br/index.php?categoria=3&pag=2&tipo=5`
- As informações no lado do servidor são recuperadas a partir do **array global \$_GET**.
- Quando o objetivo é passar parâmetros confidenciais, como exemplo as senhas, **não deve ser usado o método GET**.

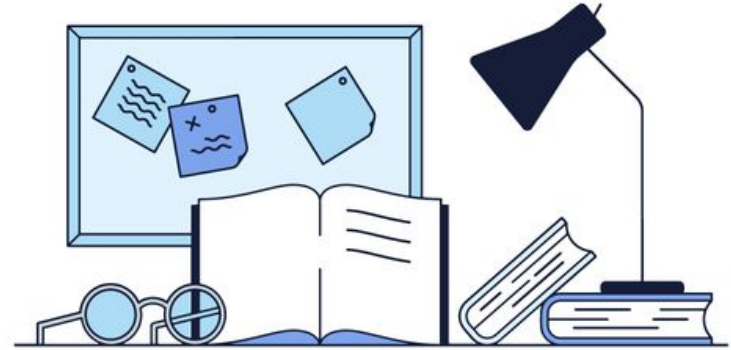


Métodos HTTP: usar GET ou POST?



- Não tem limite de capacidade para envio de informações.
- Não expõe os dados enviados na URL do navegador, pois as informações submetidas são incluídas no corpo da mensagem.
- As informações no lado do servidor são recuperadas a partir do **array global \$_POST**.

Fontes e apontamentos da aula:



<https://github.com/adrielsales/senac/wiki/Aulas-Senac>