



Formulários

- Composição de um envio de formulário
 - URL Request (url de destino que pode conter parâmetros)
 - POST (conjunto de campos e valores separado da request)
 - GET (conjunto de campos e valores integrada a request)
 - Botões não são enviados

Formulário

```
<FORM action="arquivo.php?act=save" method="POST">
Nome: <INPUT type=TEXT name=NOME><BR>
Idade: <INPUT type=TEXT name=IDADE><BR>
E-mail: <INPUT type=TEXT name=EMAIL>

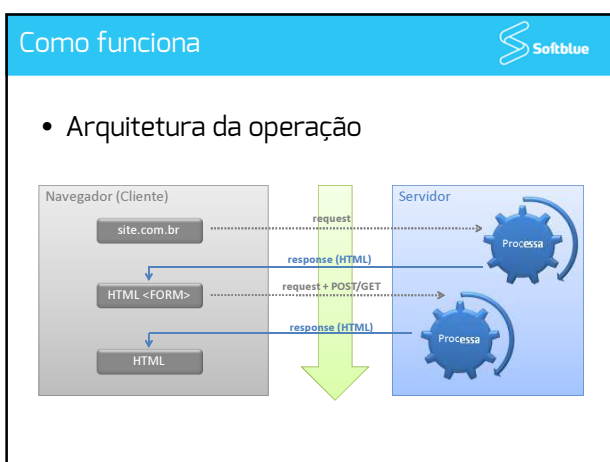
<INPUT type=RESET value="Limpar">
<INPUT type=SUBMIT value="Enviar">
</FORM>
```

Request

Request / GET

POST

NOME=André
IDADE=28
EMAIL=a@a.com



Request no Php



- Array com os valores passados via request e get

Sintaxe

`$_REQUEST`

Exemplo (<FORM action="arquivo.php?act=save">)

```
echo $_REQUEST["act"];
// Resultado: save
```

Importante

É possível alterar valores das posições do array, porém este valor não é retornado para o usuário de forma automática, mas sim, utilizado apenas no processamento da request

Post no Php



- Array com os valores passados via POST

Sintaxe

`$_POST`

Exemplo

```
echo $_POST["NOME"]; // Resultado: André
echo $_POST["IDADE"]; // Resultado: 28
echo $_POST["EMAIL"]; // Resultado: a@a.com
```

Importante

É possível alterar valores das posições do array, porém este valor não é retornado para o usuário de forma automática, mas sim, utilizado apenas no processamento da request

Variáveis automáticas



- `register_globals`

Sintaxe

`register_globals = On/Off`

Exemplo

```
echo $NOME; // Ao invés de $_REQUEST["NOME"];
echo $IDADE; // Ao invés de $_REQUEST["IDADE"];
echo $EMAIL; // Ao invés de $_REQUEST["EMAIL"];
```

Importante

Esta funcionalidade depende do arquivo de configuração de cada servidor


Validação de dados no Php



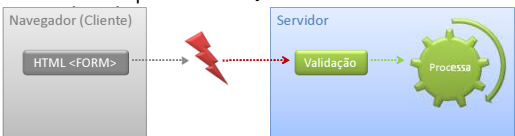
- Validação de dados no lado Cliente
 - Geralmente realizada com Javascript
 - Otimiza recursos do servidor
 - Não valida usuários mal-intencionados




Validação de dados no Php



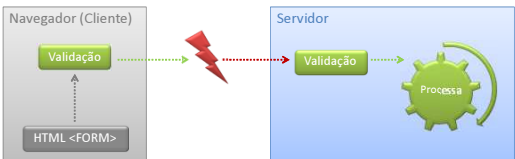
- Validação de dados no lado Servidor
 - Realizada no arquivo Php de destino do formulário
 - Aumenta o número de operações no servidor
 - Garante que a validação dos dados será



Validação de dados no Php



- Ideal: Validar em ambos os lados
 - Garante que os dados serão validados no Php
 - Poupa tráfego para usuários bem-intencionados (maioria)
 - O dobro de recursos de mão de obra são necessários



Aulas práticas e manuais on-line





Assista agora as aulas práticas, que apresentam o uso dos comandos abordados nesta aula teórica.

[Clique aqui](#) para visualizar as aulas práticas disponíveis
