

# Prof. Esp. Thalles Canela

- **Graduado:** Sistemas de Informação - Wyden Facimp
- **Pós-graduado:** Segurança em redes de computadores - Wyden Facimp
- **Consultor de Tecnologia - [aXR6] Cyber Security e NtecSoftware**
- **Professor no Senac (contratado)**
- **Professor na Wyden Facimp (contratado)**
  - **Pós-graduação:** Segurança em redes de computadores - Wyden Facimp
- **Professor na Wyden Facimp (Efetivado)**
  - **Graduação:** Análise e desenvolvimento de sistemas - Wyden Facimp

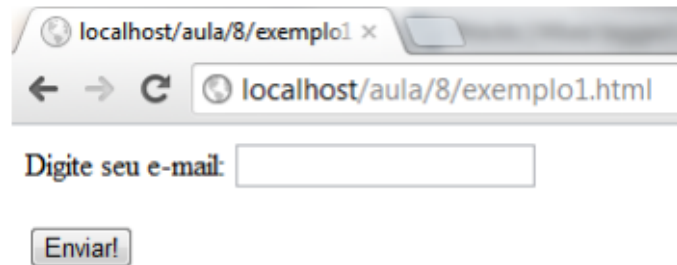
## Redes sociais:

- **Linkedin:** <https://www.linkedin.com/in/thalles-canela/>
- **YouTube:** <https://www.youtube.com/aXR6CyberSecurity>
- **Facebook:** <https://www.facebook.com/axr6PenTest>
- **Instagram:** [https://www.instagram.com/thalles\\_canela](https://www.instagram.com/thalles_canela)
- **Github:** <https://github.com/ThallesCanela>
- **Github:** <https://github.com/aXR6>
- **Twitter:** <https://twitter.com/Axr6S>

# Roteiro

- ▶ Como criar um formulário
- ▶ Enviando as informações para um programa PHP
  - Método GET
  - Método POST
- ▶ Como tratar as informações recebidas
- ▶ Funções especiais para formatação de dados
- ▶ Verificando os campos de um formulário

# Como criar um formulario



```
<form =>  
  <p>Digite seu e-mail: <input type="text" name="email" size="20"></p>  
  <p><input type="submit" value="Enviar!" name="enviar"></p>  
</form>
```

- ▶ As informações vão ser perdidas, pois o navegador não sabe o que fazer com elas
- ▶ Para o tornar útil podemos usar a opção *action*, informando ao navegador para aonde enviar a informações para serem processadas

```
<form action="recebe_dados.php">  
  <p>Digite seu e-mail: <input type="text" name="email" size="20"></p>  
  <p><input type="submit" value="Enviar!" name="enviar"></p>  
</form>
```



# TAG input

- ▶ Define um campo de entrada
- ▶ Acompanhado de diversas opções

Opção	Descrição
name	Informa qual o nome do campo
value	Informa um valor padrão para o campo
size	Informa o tamanho do campo exibido na tela
maxlength	Informa o número máximo de caracteres do campo
type	Informa qual é o tipo do campo de entrada de dados

# TAG input

## ► Os possíveis valores da opção *type*

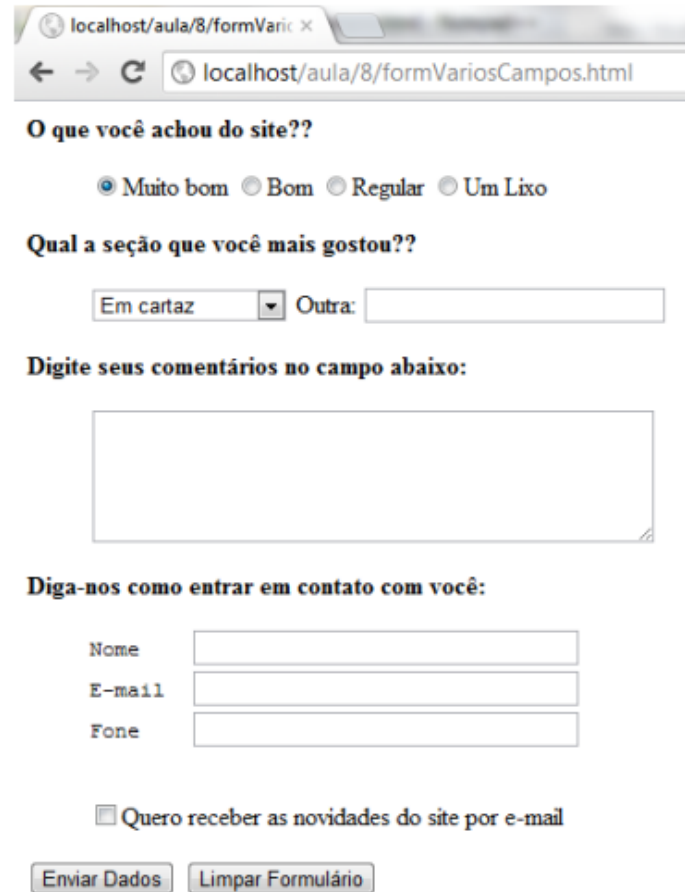
Valor	Descrição
text	Mostra uma caixa de texto de uma linha que permite a entrada de valores numéricos ou alfanuméricos
password	Usado na digitação de senhas, pois camufla qualquer caractere digitado com (*) mas a informação é enviada normalmente
hidden	Campo escondido, não aparece na tela. Usado para passar informações aos programas que recebem os dados. Muito útil
select	Uma lista de seleção (ou <i>drop-down</i> )
checkbox	Uma caixa de seleção, que pode ser marcada ou desmarcada
radio	Botões de seleção, em que só uma opção é escolhida entre várias

# TAG input

## ► Os possíveis valores da opção *type*

Valor	Descrição (Continuação)
textarea	Caixa de texto com várias linhas
file	Permite o envio de arquivos
submit	Botão que aciona o envio dos dados dos formulários
image	Mesma função <i>submit</i> , mas uma imagem substitue o botão tradicional
reset	Limpa todos os campos de um formulário e retorna o valor-padrão (se existir)

# Como criar um formulario



localhost/aula/8/formVariosCampos.html

O que você achou do site??

☒ Muito bom ☐ Bom ☐ Regular ☐ Um Lixo

Qual a seção que você mais gostou??

Em cartaz  Outra:

Digite seus comentários no campo abaixo:

Diga-nos como entrar em contato com você:

Nome

E-mail

Fone

☐ Quero receber as novidades do site por e-mail

# Como criar um formulario

```
<form action="processa.php">
  <p>
    <strong>O que você achou do site??</strong>
    <dl><dd>
      <input type="radio" name="avaliacao" value="muitobom" checked> Muito bom
      <input type="radio" name="avaliacao" value="bom"> Bom
      <input type="radio" name="avaliacao" value="regular"> Regular
      <input type="radio" name="avaliacao" value="umlixo"> Um Lixo
    </dd></dl>
  </p>
  <p>
    <strong>Qual a seção que você mais gostou??</strong>
    <dl><dd>
      <select name="secao" size="1">
        <option value="emcartaz"> Em cartaz</option>
        <option value="trilhasonora"> Trilha Sonora</option>
        <option value="fotos"> Galeria de Fotos</option>
        <option value="bilheteria"> Bilheteria</option>
        <option value="outra"> Outra</option>
      </select>
      Outra: <input type="text" size="26" maxlength="256" name="outra">
    </dd></dl>
  </p>
```



# Como criar um formulario

```
<p>
    <strong>Digite seus comentários no campo abaixo:</strong>
    <dl><dd>
        <textarea name="comentarios" rows="5" cols="42"></textarea>
    </dd></dl>
</p>
<p>
    <strong>Diga-nos como entrar em contato com você:</strong>
    <dl><dd>
        <pre>
Nome <input type="text" size="35" maxlength="256" name="nome">
E-mail <input type="text" size="35" maxlength="256" name="email">
Fone <input type="text" size="35" maxlength="256" name="fone">
        </pre>
    </dd></dl>
    <dl><dd>
        <input type="checkbox" name="novidades" value="nov">Quero receber as novidades do site por e-mail
    </dd></dl>
</p>
<p>
    <input type="submit" value="Enviar Dados">
    <input type="reset" value="Limpar Formulário">
</p>
</form>
```

# Enviando informações em PHP

- ▶ Utilizamos a opção da *action* da tag *form* do HTML
  - `<form action="processa.php">`
- ▶ Existem dois métodos de passagem de parâmetros
  - GET
  - POST
- `<form action="processa.php" method="POST">`

# GET

- ▶ Método padrão de envio de dados
  - Se não for especificado o método na tag *action*, GET é assumido pelo PHP
- ▶ Os dados são enviados juntamente com o nome da página (URL) para o envio de dados

```
<form action="recebe_dados.php">  
    <p>Digite seu nome: <input type="text" name="nome" size="30"></p>  
    <p>Digite seu e-mail: <input type="text" name="idade" size="3"></p>  
    <p><input type="submit" value="Enviar!" name="enviar"></p>  
</form>
```

- ▶ [http://www.seusite.com.br/recebe\\_dados.php?nome=Joaquim&idade=20](http://www.seusite.com.br/recebe_dados.php?nome=Joaquim&idade=20)
  - ? – representa o início da cadeia de variáveis
  - & – identifica o início de uma nova variável
  - = – separa as variáveis dos seus respectivos valores



# GET

## ► Desvantagens

- Limite de caracteres é de 2.000
- Os dados enviados são visíveis na barra de endereço do navegador
- O método POST resolve isso

## ► Vantagem

- Pode ser utilizado para passagem de parâmetros por link

# GET

- ▶ Vantagem

- Um loja virtual com um link para cada produto em que o identificador é passado como parâmetro  
Podem ser passados mais de uma parâmetro, como uma subcategoria

- ▶ `http://www.sualojinha.com.br/produto.php?id_produto=23&sub_categoria=8`

- ▶ Recebidas os parâmetros, um programa processa os dados carregando as informações do banco de dados

# POST

- Basta usar o método POST em tag *form*

```
<form action="recebe_dados.php" method="POST">  
  <p>Digite seu nome: <input type="text" name="nome" size="30"></p>  
  <p>Digite seu e-mail: <input type="text" name="idade" size="3"></p>  
  <p><input type="submit" value="Enviar!" name="enviar"></p>  
</form>
```

- Diferente do GET, o POST envia os dados por meio do corpo da mensagem encaminhada ao servidor

# POST

## ► Vantagens

- Não é visível a cadeia de variáveis
  - `http://www.seusite.com.br/recebe_dados.php`
- Não limites no tamanho dos dados, sendo mais usado para formulários com grande quantidade de informações
- Enviar outros tipos de dados, não aceitos pelo GET, como imagens ou outros arquivos (usar valor *file* na opção *type* da tag *input*)

## ► Desvantagens

- Não é possível a passagem de parâmetros

# Como tratar as informações recebidas

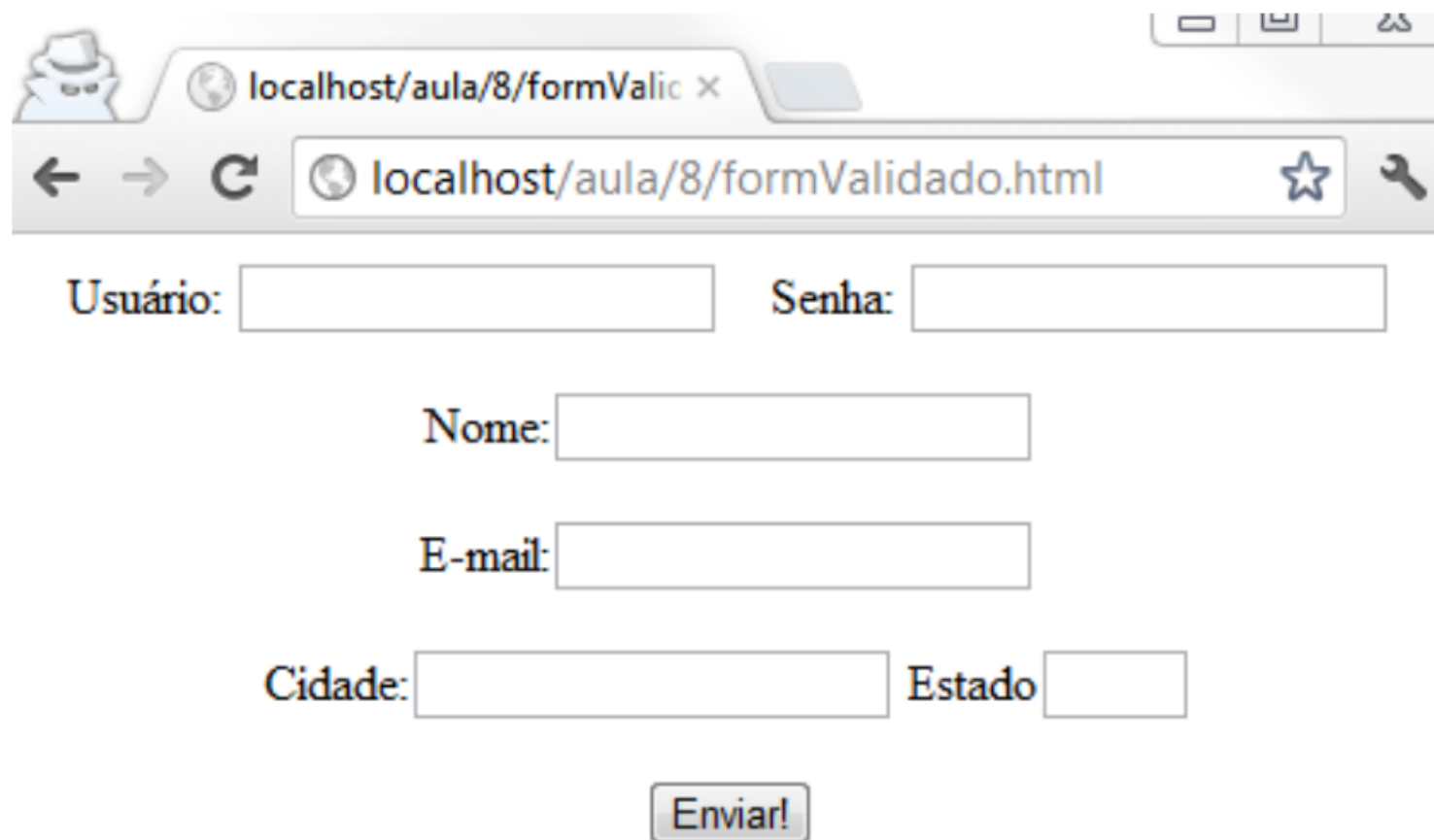
- ▶ **Existem duas maneiras de acessar os dados recebidos**
  - Tratar como variáveis adicionando \$ ao nome dos campo especificados no formulário. O campo *nome* ficaria *\$nome* e o campo *idade* *\$idade*.
    - É necessário que a opção *register\_globals* seja ativada
  - Usar os arrays superglobais predefinidos pelo php: `$_GET` e `$_POST`
    - Os nome dos campos são usado como chaves associativas `$_GET["nome"]` ou `$_POST["idade"]`
- ▶ Por questões de segurança os desenvolvedores do PHP recomendam o uso do arrays `$_GET` e `$_POST`. Caso queira usar a primeira opção, cuide para não criar outras variáveis com o mesmo nome, pois o valores serão sobrescritos



# Funções especiais para formação de dados

- ▶ Outras duas funções importante ao usar o método GET são **urldecode** e **urlencode**
  - Usando GET “Joaquim Legal” fica  
`http://www.seusite.com.br/recebe_dados.php?nome=Joaquim%20Legal`
  - Para ler corretamente
    - `$nome = urldecode($nome);`
- ▶ A função **urldecode** retira os códigos hexadecimais e **urlencode** recoloca
- ▶ **Urlencode** é útil para redirecionar as informações para outro local por meio do método GET

# Verificando um formulário



A screenshot of a web browser window. The address bar shows the URL `localhost/aula/8/formValidado.html`. The page contains a form with the following fields and labels:

- Usuário:**
- Senha:**
- Nome:**
- E-mail:**
- Cidade:**
- Estado:**

At the bottom center of the form is a button labeled **Enviar!**.

# Verificando um formulário

```
<form action="recebe_dados.php" method="POST">  
    <div align="center"><center>  
        <p>Usuário: <input type="text" name="usuario" size="20">     </p>  
  
        Senha: <input type="text" name="senha" size="20"></p>  
        <p>Nome:<input type="text" name="nome" size="20"></p>  
        <p>E-mail:<input type="text" name="email" size="20"></p>  
        <p>Cidade:<input type="text" name="cidade" size="20">  
Estado<input type="text" name="estado" size="2" maxlength="2"></p>  
  
        <p><input type="submit" value="Enviar!" name="enviar"></p>  
    </center></div>  
</form>
```

# Verificando um formulário

- Recebe\_dados.php – Recebe e processa os dados da página

```
$usuario = $_POST["usuario"];
$senha = $_POST["senha"];
$nome = $_POST["nome"];
$email = $_POST["email"];
$cidade = $_POST["cidade"];
$estado = $_POST["estado"];
$erro = FALSE;

if(strlen($usuario)<5)
{echo "O usuário deve possuir ao menos 5 caracteres. <br>";          $erro=TRUE;}
if(strlen($senha)<5)
{echo "A senha deve possuir ao menos 5 caracteres. <br>";          $erro=TRUE;}
if(empty($nome) OR strpos($nome,' '))
{echo "Digite o nome corretamente. <br>";                          $erro=TRUE;}
if(strlen($email)<8 OR !strpos($email,'@'))
{echo "Digite o e-mail corretamente. <br>";                          $erro=TRUE;}
if(empty($cidade))
{echo "Digite a cidade corretamente. <br>";                          $erro=TRUE;}
if(strlen($estado)!=2)
{echo "Digite o estado corretamente. <br>";                          $erro=TRUE;}
if(!$erro)
{echo "Todos os dados foram digitados corretamente! <br>";}
```

# Verificando um formulário

Função	Descrição
<code>empty</code>	Verifica se a string está vazia
<code>strlen</code>	Retorna o número de caracteres de uma string
<code>strstr</code>	Acha a primeira ocorrência de um caractere em uma string