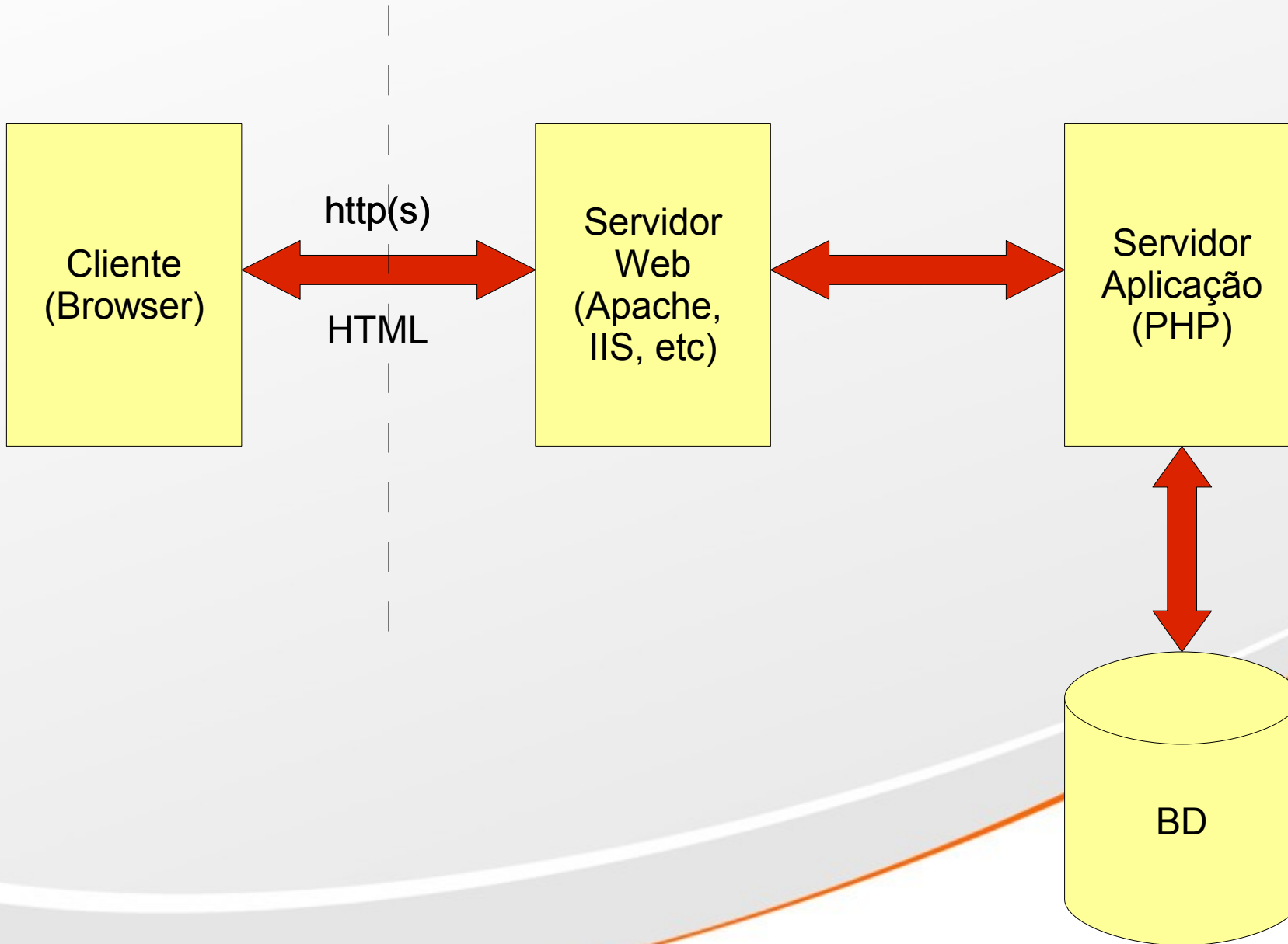


# Programação web

# Arquitetura Web



# Requisições



# Formulário GET

```
<form action="index.php" method="GET">
```

```
List: <input type="text" name="list" /><br />
```

Order by:

```
<select name="orderby">
```

```
<option value="name">Name</option>
```

```
<option value="city">City</option>
```

```
<option value="zip">ZIP Code</option>
```

```
</select><br />
```

Sort order:

```
<select name="direction">
```

```
<option value="asc">Ascending</option>
```

```
<option value="desc">Descending</option>
```

```
</select>
```

```
</form>
```

# GET e URLs

`http://example.org/index.php?list=user&orderby=name&direction=asc`

```
echo $_GET['list'];
```

`http://example.org/index.php?list=user&order[by]=column&order[dir]=asc`

```
echo $_GET['order']['by'];
```

```
echo $_GET['order']['direction'];
```

```
$data = "Max & Ruby";
```

```
echo "http://www.phparch.com/index.php?name=" . urlencode ($data);
```

# Formulário POST

```
<form action="index.php" method="POST">  
    <input type="hidden" name="login" value="1" />  
    <input type="text" name="user" />  
    <input type="password" name="pass" />  
</form>
```

```
if ($_POST['login']) {  
    if ($_POST['user'] == "admin" &&  
        $_POST['pass'] == "secretpassword") {  
    }  
}
```

# Formulário POST

```
<form method="post">

  <p>Escolha as linguagens que você tem conhecimento:</p>

  <p>

    <label><input type="checkbox" name="linguagens[]"
    value="PHP" /> PHP</label> <br />

    <label> <input type="checkbox" name="linguagens[]"
    value="Perl" /> Perl </label> <br />

    <label> <input type="checkbox" name="linguagens[]"
    value="Ruby" /> Ruby </label>

    <br />

    <input type="submit" value="Enviar" name="botao" />

  </p>

</form>
```

# Formulário POST

```
foreach ($_POST['linguagens'] as $linguagem) {  
    switch ($linguagem) {  
        case 'PHP' :  
            echo "PHP? Excelente! <br />";  
            break;  
        case 'Perl' :  
            echo "Perl? Legal. <br />";  
            break;  
        case 'Ruby' :  
            echo "Ruby? Boa sorte. <br />";  
            break;  
        default:  
            echo "Linguagem desconhecida!";  
    }  
}
```



# Tratando GET e POST juntos

- Se não certeza se o formulário será enviado via GET ou POST, pode-se utilizar a global `$_REQUEST` que recebe dados do GET e do POST.
- Não há como saber se o dado foi enviado via GET ou POST.
- Um problema pode ser criado na utilização do `$_REQUEST` em relação a segurança, facilitando que um formulário de seu sistema tenha seu facilmente simulado.

# UPLOAD

```
<form enctype="multipart/form-data" action="index.php"
  method="post">

  <input type="hidden" name="MAX_FILE_SIZE" value="50000" />

  <input name="arquivo" type="file" />

  <input type="submit" value="Send file" />

</form>
```

- O valor de MAX\_FILE\_SIZE é um aviso para o browser. É fácil contornar este limite. Então não conte que o browser irá obedecer a sua vontade. O que foi estabelecido para maximum-size no PHP não pode ser enganado. Mas você deve adicionar MAX\_FILE\_SIZE em qualquer caso, já que salva os usuários do problema de esperar por um grande arquivo ser transferido somente para descobrir depois de tudo que ele é muito grande.

# Dados do upload

- `$_FILES['arquivo']['name']`
  - O nome original do arquivo no computador do usuário.
- `$_FILES['arquivo']['type']`
  - O tipo mime do arquivo, se o browser deu esta informação. Um exemplo pode ser "image/gif".
- `$_FILES['arquivo']['size']`
  - O tamanho, em bytes, do arquivo.
- `$_FILES['arquivo']['tmp_name']`
  - O nome temporário do arquivo, como foi guardado no servidor.
- `$_FILES['arquivo']['error']`
  - O código de erro associado a este upload de arquivo. ['error'] foi adicionado no PHP 4.2.0

# Tratando upload

```
$uploadaddir = '/var/www/uploads/';

$uploadfile = $uploadaddir . $_FILES['arquivo']['name'];

print "<pre>";

if (move_uploaded_file($_FILES['arquivo']['tmp_name'], $uploadaddir
. $_FILES['arquivo']['name'])) {

    print "O arquivo é valido e foi carregado com sucesso. Aqui
esta alguma informação:\n";

    print_r($_FILES);
} else {

    print "Possivel ataque de upload! Aqui esta alguma
informação:\n";

    print_r($_FILES);
}
```

# Erros de upload

- `UPLOAD_ERR_OK`
  - Valor: 0; não houve erro, o upload foi bem sucedido.
- `UPLOAD_ERR_INI_SIZE`
  - Valor 1; O arquivo no upload é maior do que o limite definido em `upload_max_filesize` no `php.ini`.
- `UPLOAD_ERR_FORM_SIZE`
  - Valor: 2; O arquivo ultrapassa o limite de tamanho em `MAX_FILE_SIZE` que foi especificado no formulário HTML.
- `UPLOAD_ERR_PARTIAL`
  - Valor: 3; o upload do arquivo foi feito parcialmente.
- `UPLOAD_ERR_NO_FILE`
  - Valor: 4; Não foi feito o upload do arquivo.

# Configurações específicas

- Diretivas
  - post\_max\_size
  - max\_input\_time
  - upload\_max\_filesize
  - memory\_limit

# Cabeçalhos HTTP

```
header("Location: http://phparch.com");
```

```
exit();
```

```
ob_start("ob_gzhandler");
```

```
header("Cache-Control: no-cache, must-revalidate");
```

```
header("Expires: Thu, 31 May 1984 04:35:00 GMT");
```

```
$date = gmdate("D, j M Y H:i:s", time() + 2592000); // 30 Days  
from now
```

```
header("Expires: " . $date . " UTC");
```

```
header("Cache-Control: Public");
```

```
header("Pragma: Public");
```

- `zlib.output_compression = on`
- `zlib.output_compression_level = 9`

# Cookie

```
setcookie("hide_menu", "1");
```

```
setcookie("hide_menu", "1", time() + 86400);
```

```
if ($_COOKIE['hide_menu'] == 1) {  
    // esconder um menu  
}
```

```
setcookie("test_cookie[0]", "foo");
```

```
setcookie("test_cookie[1]", "bar");
```

```
setcookie("test_cookie[2]", "bar");
```

```
setcookie("hide_menu", false, -3600); // Apagar cookie
```



# Sessão

- Quando um site é acessado, o PHP irá conferir automaticamente (se `session.auto_start` estiver definido como 1) ou quando você pedir (explicitamente através de `session_start()` ou implicitamente através de `session_register()`) quando um id de sessão específico for enviado com a requisição. Se este for o caso, o ambiente anteriormente salvo é recriado.

# Sessões – diretivas de configurações

- `session.save_handler`
- `session.save_path`
- `session.name`
- `session.use_cookies`
- `session.cache_expire`

# Sessão

```
session_start();  
  
// Com PHP menor ou igual a 4.0.6 usar $HTTP_SESSION_VARS  
  
if (!isset($_SESSION['count'])) {  
    $_SESSION['count'] = 0;  
} else {  
    $_SESSION['count']++;  
}
```

# Sessão

- session\_start
- session\_destroy
- session\_id
- session\_name
- session\_save\_path
- session\_regenerate\_id