Páginas Dinâmicas

As páginas WWW estáticas são documentos em formato HTML armazenados no servidor, como esta página. A estrutura básica de uma página HTML segue o formato do exemplo a seguir:

```
<HTML>

<HEAD>
<TITLE>
Este &eacute; o t&iacute;tulo desta p&aacute;gina
</TITLE>
</HEAD>

<BODY>
Este &eacute; o conte&uacute;do da p&aacute;gina,
que pode ter diversas formata&ccedil;&otilde;es.
</BODY>

</HTML>
```

As páginas estáticas podem conter imagens, sons, animações e outros elementos. Todavia, como seu próprio nome diz, seu conteúdo de informação é estático e sua atualização depende de intervenção manual. Um sistema assim pode ser considerado um sistema *two-tiers* de informação estática, pois o servidor WWW funciona basicamente como um repositório de informação estática.

A seguir veremos mecanismos que permitem gerar páginas com conteúdo dinâmico, ou seja, que pode mudar automaticamente. Nesse caso, o sistema passa a ser caracterizado com um sistema *three-tiers*, no qual o servidor WWW atua como um elemento intermediário entre o cliente e o repositório de dados.

Programas CGI

O método mais antigo e simples para prover informação dinâmica na WWW é o uso do mecanismo de programas CGI - *Common Gateway Interface*. A tecnologia CGI permite a construção sob demanda de páginas WWW com conteúdo variável. Ela se baseia na solicitação de execução de um programa ou script no servidor, que irá gerar uma saída compreensível para o navegador no lado do cliente. Um programa CGI é um arquivo instalado no servidor que, ao ser acessado por um cliente, é executado e devolve ao cliente o resultado de sua execução.

A configuração default do servidor Apache somente permite a execução de CGIs presentes no diretório `/var/www/cgi-bin`. Essa configuração deve ser alterada caso se deseje habilitar a execução de CGIs em outros diretórios, como por exemplo os diretórios `$HOME/public_html` dos usuários. Todavia isso deve ser feito com cuidado, para não criar brechas na segurança do sistema.

Um exemplo bastante simples de programa CGI (escrito em shell script) pode ser visto aqui:

```
#!/bin/sh

echo "Content-type: text/plain"
echo ""

echo -n "Data atual neste servidor: "
date
```

Aqui temos outro exemplo mais elaborado, gerando como saída uma página em HTML:

```
#!/bin/sh

echo "Content-type: text/html"
echo ""

echo "<HTML>"
echo "<HEAD>"
echo "<TITLE>Uma página simples em CGI shell</TITLE>"
echo "</HEAD>"

echo "<BODY>"

echo "As variáveis de ambiente estão acessíveis ao usuário:"

echo "<UL>"
echo "<LI>Bem-vindo, usuário em $REMOTE_ADDR</LI>"
echo "<LI>Você está usando o browser $HTTP_USER_AGENT</LI>"
echo "<LI>Este servidor executa em $SERVER_SOFTWARE</LI>"
echo "</UL>"

echo "Esta abordagem somente deve ser usada em <b>páginas muito simples</b>."

echo "</BODY>"
echo "</HTML>"
```

É possível a criação de CGIs simples com scripts shell, ou mesmo com programas C/C++, mas essas linguagens não se prestam à construção de CGIs que devam retornar código HTML, pois a construção do mesmo via shell scripts ou programas C é penosa e pouco eficiente. Para tal objetivo costumam ser usadas linguagens apropriadas para tratamento de strings, como Perl ou Python.

Eis um exemplo de programa CGI em Perl. Ele gera uma tabela com informações dos discos locais do servidor no instante da consulta:

```
#!/usr/bin/perl

# cabecalho HTTP
print "Content-type: text/html\n\n";

# inicio do documento HTML
print "<HTML>
<HEAD> <TITLE>Discos locais</TITLE> </HEAD>

<BODY>
<H1>Situação dos discos locais</H1>
";

# obtem a situação dos discos locais
@lista = `df` ;

# abertura de tabela
print "<TABLE COLS=5 BORDER=2>\n" ;

# gera entradas de tabela para as linhas
foreach $line ( @lista )
```

```
{
  ($fs, $bks, $used, $avail, $perc, $mount) = split ( /\s+/, $line) ;
  print "<TR>
    <TD>$fs</TD>
    <TD>$bks</TD>
    <TD>$used</TD>
    <TD>$avail</TD>
    <TD>$perc</TD>
    <TD>$mount</TD>
    </TR>
    ";
}

# fechamento de tabela
print "</TABLE>\n";

# fim do documento HTML
print "</BODY></HTML>\n";
```

Resumindo, programas CGI são códigos executáveis lançados pelo servidor WWW como processos separados, cujo resultado da execução (saídas geradas em `stdout`) são enviadas de volta ao navegador.

Páginas ativas

Outra forma de gerar conteúdo dinâmico na Web é através do uso de páginas ativas. Neste caso, o documento HTML contém tags especiais definindo instruções executáveis. Essas instruções são executadas pelo servidor WWW no momento da consulta, e o resultado da execução é inserido na página Web e devolvido ao cliente. Atualmente as linguagens para páginas ativas em uso mais freqüentes são:

- PHP: de domínio públio, executa em qualquer plataforma.
- ASP: proprietária, executa sobre servidores Windows.
- JSP: baseada em Java, pode executar na maioria dos ambientes.

A discussão sobre as qualidades e defeitos dessas linguagens foge ao escopo desta aula. Além das duas linguagens acima, existem outras possibilidades, como Embedded Perl, Python, etc.

Um exemplo simples de página em PHP seria o seguinte:

```
<HTML>
<BODY>

<?
   $IP = getenv("REMOTE_ADDR");
   echo "Olá usuário em $IP, tudo bem?" ;
?>

</BODY>
</HTML>
```

O código indicado em azul é identificado por tags especiais (<? e ?>) e executado pelo servidor ao carregar a página. O resultado da execução, sem o código, é retornado ao navegador do cliente. Vejamos outro exemplo mais complexo, no qual uma imagem é gerada dinamicamente (para observar melhor seu funcionamento, recarregue a página várias vezes, através do botão *reload* de seu navegador):

```
<?
   // Exemplo de criação de imagem

   // aloca uma área de memória para uma imagem 200x200
   $im = ImageCreate (200, 200);

   // define cores para usar na imagem
   $black = ImageColorAllocate ($im, 0, 0, 0) ;
   $red   = ImageColorAllocate ($im, 255, 0, 0) ;
   $green = ImageColorAllocate ($im, 0, 255, 0) ;
   $blue  = ImageColorAllocate ($im, 0, 0, 255) ;

   // Desenha alguns retangulos aleatorios na tela
   srand ((double) microtime() * 1000000);

   $x = rand (0,100) ;
   $y = rand (0,100) ;
   ImageFilledRectangle ($im, $x, $y, $x+100, $y+100, $red);

   $x = rand (0,100) ;
   $y = rand (0,100) ;
   ImageFilledRectangle ($im, $x, $y, $x+100, $y+100, $green);

   $x = rand (0,100) ;
   $y = rand (0,100) ;
   ImageFilledRectangle ($im, $x, $y, $x+100, $y+100, $blue);

   // cria um arquivo em disco com a imagem "imagem.png"
   ImagePng ($im, "imagem.png");

   // desaloca a memória utilizada para manipular a imagem
   ImageDestroy ($im);
?>

<html>

<head>
<title>PHP Exemplos - Gr&aacute;ficos</title>
</head>

<body bgcolor = "#FFFFFF">

<img align="center" src="imagem.png">
<p>Recarregue a p&aacute;gina para mudar a imagem</p>
</body>
</html>
```

Outro exemplo de geração de conteúdo dinâmico é esta página, que gera a imagem de um dado com um valor aleatório cada vez que é acessada. Ela poderia servir, por exemplo, para jogar General!

Ao invés de gerar imagens, o código PHP poderia buscar informações em um servidor de bancos de dados. Dessa forma, o código PHP pode ser considerado o “middle-tier” de um sistema de informação, que mantém a lógica do negócio. Um exemplo desse funcionamento pode ser observado no script PHP abaixo:

```
<HTML>
<BODY>
```

```
<?
  // variáveis usadas na conexão ao DBMS
  $host  = "espec.ppgia.pucpr.br";
  $user  = "xxxxx";
  $pwd   = "xxxxx";
  $base  = "cadastro";
  $table = "usuarios";

  // estabelecer conexão com o DBMS
  mysql_connect($host, $user, $pwd)
     OR die ("Falhou no acesso ao DBMS");

  // selecionar uma base de dados
  mysql_select_db( "$base")
     OR die( "Falhou na seleção da base");

  // seleciona todos os usuários cadastrados
  $query = "SELECT * FROM $table";
  $result = mysql_query($query);

  // quantas entradas temos no resultado ?
  $number = mysql_numrows($result);

  // gerar uma saida formatada
  if ($number == 0) :
?>

<CENTER><P>Cadastro vazio !</CENTER>

<? else : ?>

<H3>Usuários cadastrados: <? echo $number ?></H3>

<table border=1>
<tr bgcolor="#AAAAAA">
<td>Nome</td> <td>Endereço</td> <td>Cidade</td> <td>Estado</td>
<td>E-mail</td> </tr>

<?  for ($i=0; $i < $number; $i++):
      $reg = mysql_fetch_array ($result) ;
?>

<tr>
<td> <? echo $reg["nome"]   ?> </td> <td> <? echo $reg["ender"]  ?> </td>
<td> <? echo $reg["cidade"] ?> </td> <td> <? echo $reg["estado"] ?> </td>
<td> <? echo $reg["email"]  ?> </td>
</tr>

<?  endfor;
?>

</table>

<? endif;
?>
```

```
</BODY>
</HTML>
```

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=espec:paginas_dinamicas

Last update: **2012/04/25 19:42**