CENTRO PAULA SOUZA

GOVERNO DO ESTADO DE
SÃO PAULO

www.centropaulasouza.sp.gov.br

Linguagem PHP

Exemplo de uma Aplicação

Plano Geral

- Histórico
- Definições e Características
- O primeiro programa
  - onde gravar na estrutura CGI?
  - As variáveis PHP: Tipos de Dados e propriedades
- Estrutura de Execução de Programas
  - Processos / Funções e subprogramas
  - Estruturas de Desvio de Fluxo de execução
- Categorias de Programas ←→ Modelo de Dados
- Construindo um programa recursivo
  - Primeiro entenda o separado... Depois coloque num só arquivo
- Montando um programa que emite HTML
  - FRAMES e MENUS
- Construindo um ICEAL básico
  - Lendo dados
  - Tratamento de Transação
    - Inicio / Abandono / Conclusão
  - Funções:
    - Acessando o Banco de Dados
    - Caixa de Seleção de Uma FK
    - Outras
    - Montando o pool de funções

PHP - Histórico

- 1994 Rasmus Lerdorf escreve um interpretador de comandos para montagem de homepages personalizadas (Personal HomePage).
- 1997 - 50.000 sites
- outubro de 1998 - 100.000 sites
- 1999 -1.000.000 sites
- Zeev Suraski e Andi Gutmans, dois programadores israelenses que desenvolveram os analisadores de sintaxe PHP3 e PHP4.
- A empresa ZEND passa a coordenar o desenvolvimento da Linguagem PHP
- Acabou o histórico.

PHP – Definições e Características

- O que é PHP?
  - PHP
    (Primeira sigla: Personal Home Page, Hoje: Hypertext PreProcessor")
  - Inicialmente usada para o desenvolvimento de aplicações Web encapsuladas dentro de códigos HTML.
  - Hoje: Linguagem de script Open Source de uso geral (CGI). Especialmente preparada para isso

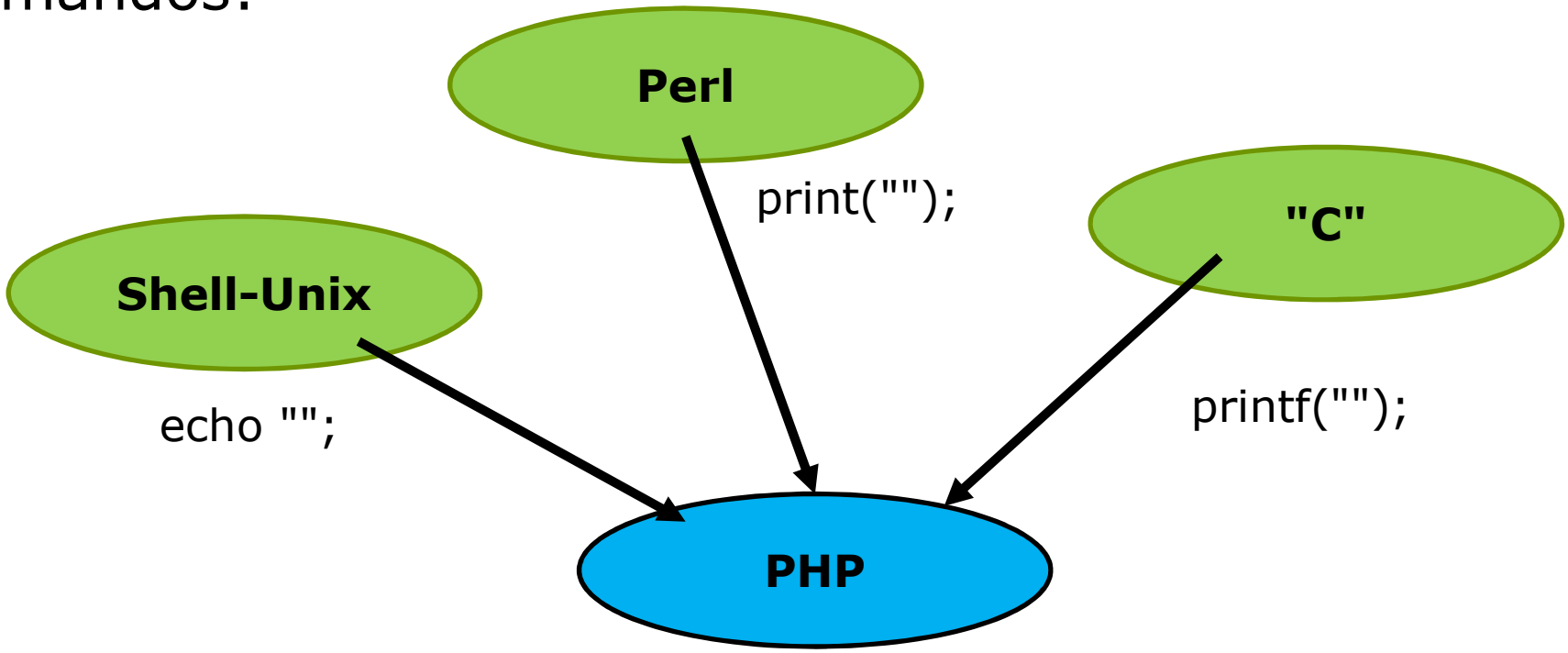
- O que o PHP pode fazer?
  - PHP não é limitado a gerar somente HTML.
  - As habilidades do PHP incluem geração de imagens, arquivos PDF e animações Flash (utilizando libswf ou Ming)
  - Você pode criar qualquer padrão texto, como XHTML e outros arquivos XML... E mais.

PHP - Definições e Características

- Suporte a um grande número de bancos de dados e acessa uma diversidade de serviços (servidores HTTP, SNMP, POP3, etc.)
- Por conta disso, é uma linguagem adequada ao desenvolvimento de sistemas na arquitetura MVC.
- É uma linguagem que permite criar 'sites dinâmicos' (páginas resultantes de acessos e processamento de dados – não ficam gravadas nos servidores).
- As páginas são “escritas” por scripts CGI, que têm a LÓGICA de uma regra de negócio (o usuário não acessa o fonte do script preservando a propriedade intelectual do autor).
- As páginas são “montadas” on-fly, ou seja em tempo de execução. As páginas não são armazenadas e sim escritas durante o processamento de um programa PHP.
  - https://www.youtube.com/watch?v=L2zqTYgcpfg

PHP – Definições e Características

- Os tipos de Arquivos (de scripts) tratados no ambiente PHP: phtml, php3 e **php**.
- PHP tem três linguagens das quais 'herdou' muitos comandos:



printf("texto $*variavel1* texto $*variável2* texto");

PHP – O primeiro programa

- Escrevendo o primeiro programa PHP
  - Edite um arquivo com um editor de programação e escreva as seguintes linhas:

```
<?php
  $CorDaCasa="Marrom";
  $Tamanho="220m2";
  printf("Olá, mundo!\n");
  printf("Moro em uma casa $CorDaCasa e quem tem $Tamanho!");
?>
```

  - O início DEVE ser com a linha <?php
  - O comando printf é IGUAL ao da linguagem C, mas também podemos escrever print (como em Perl ou echo como em shell-script).
  - Não é preciso declarar o tipo de dado de variáveis.

PHP – O primeiro programa

- Onde gravar?
  - Crie um sub diretório no diretório associado ao *localhost* no seu computador. Dê o nome de lbd.
  - Grave o arquivo com nome ***primeiroprograma.php*** no diretório onde o servidor de página acesse a execução de programas CGI.
    - Por uma situação **MUITO** conveniente o Apache deixe os programas CGI do tipo PHP no mesmo diretório onde estão os arquivos HTML.
      - É possível mudar isso alterando diretivas no arquivo httpd.conf, mas não vou ensinar isso agora.
  - Testando: abra seu navegador e na barra de endereços escreva:
    - http://localhost/lbd/primeiroprograma.php

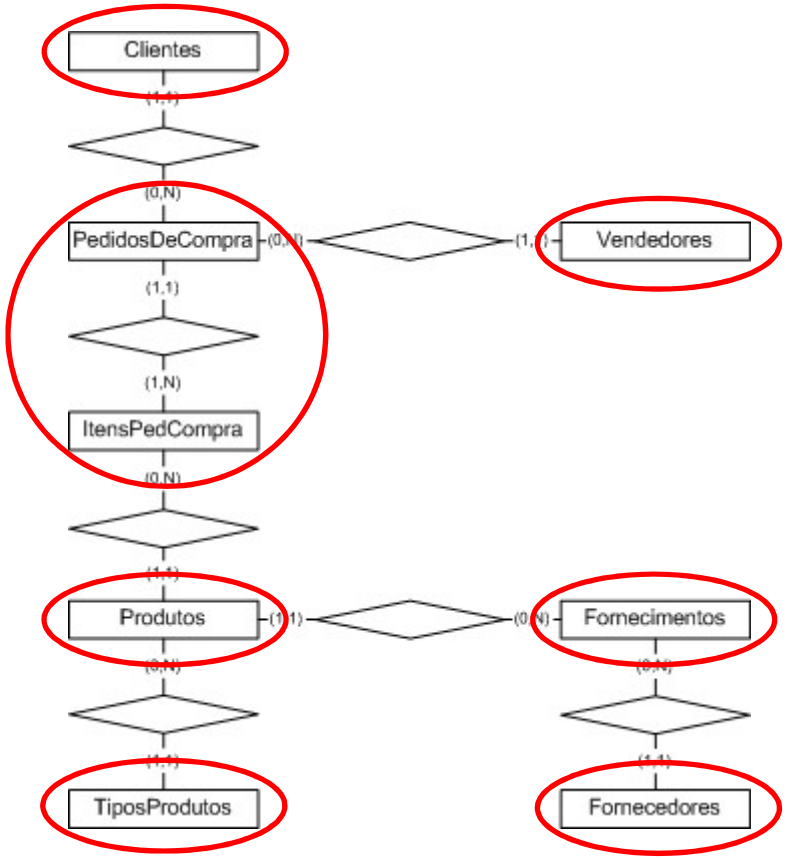
PHP – O primeiro programa

- Colocando comentários nos programas
  - Igual ao modo de comentários da linguagem C
  - Igual ao modo de comentários de Shell-Script
- Exemplo:

```
<?
  printf("teste\n"); # isto é um teste
  printf("teste\n"); // este teste é similar ao anterior
  printf("teste\n");
  /* Isto é um comentário em bloco (ou em mais de uma
     linha. Pode ser usado para documentar programas
     de modo mais completo. */
  printf("teste\n");
  #----------------------------
  # Se bem que eu prefira
  # assim: dá para formar caixas de texto mais regulares.
  #----------------------------
?>
```

PHP – Categorias de Programas em Modelos de Dados

- Categorias de Programas ←→ Modelo de Dados



A – Tabelas que NÃO tem FK

B – Tabelas que TEM pelo menos uma FK, mas as FK podem não fazer parte da sua PK

C – Tabelas que TEM pelo menos duas FK compondo sua PK

D – Tabelas onde a existência de ocorrências DEPENDE da existência de ocorrências em outras tabelas.

Categorias de programas

- Mas... E daí? O que implica em ter esta classificação?
- Bem. Se tivermos que desenvolver programas para gestão de CLIENTES estes programas serão MUITO parecidos com os programas de VENDEDORES... Em outras palavras: acelera-se o desempenho no desenvolvimento dos PAs montando TEMPLATES ou programas guia para cada categoria percebida no modelo.
- Depois de montado o template basta executar busca-e-troca para nomes de tabelas e campos... E implementar alguma regra de comportamento diferente para a tabela tratada no PA. Por exemplo se CLIENTE tiver uma regra de validação de crédito esta deve ser escrita no PA que trata CLIENTE depois de ter sido feita a busca-e-troca a partir do template.
- Podemos então pensar que o desenvolvimento pode ser guiado por 4 TEMPLATES.

PHP – Estrutura Básica de um PA

- As parte básicas de um PA
  - Coletar DADOS
  - Acessar os bancos de dados
  - Disparar a transação do PA
    - Controlar o andamento da transação
      - Tratar as situações de erro ou
      - Concluir a transação
  - Avisar o usuário que executa o PA o que ocorreu durante a execução.
- Os PAs farão basicamente 5 ações:
- Incluir, Consultar, Alterar, Excluir e Listar dados de uma tabela (**ICAEL**).

PHP – Estrutura Básica de um PA

- Um programa que monta um form e aciona outro programa.

form.php

```
<html>
<body>
<form action='p2.php' method='POST'>
Campo N1: <input type='text' name='n1' size=4 maxlength=4>
<input type='submit' value='manda'>
</form>
</body>
</html>
```

- É código HTML escrito dentro de arquivo .php. Isso *pode* não ser recomendável (pode sobrecarregar a troca de mensagens entre o interpretador PHP e o servidor de página – e vice-versa), para evitar qualquer problema... escrevemos o comando printf emitindo estas TAGS HTML. O código fica assim...

PHP – Estrutura Básica de um PA

- Um programa que monta um form e aciona outro programa.

```
<?php
printf("<html>\n");
printf("<body>\n");
printf("<form action='p2.php' method='POST'>\n");
printf("Campo N1: <input type='text' name='n1' size=4 maxlength=4>\n");
printf("<input type='submit' value='manda'>\n");
printf("</form>\n");
printf("</body>\n");
printf("</html>\n");
?>
```

- Como o interpretador deve mudar agora colocamos o <?php e ?> no arquivo.
  NOTE: o tipo do arquivo (terminação .php no nome) NÃO deve mudar.
- Note, como só temos o comando printf();, o mesmo programa pode ser escrito assim...

PHP – Estrutura Básica de um PA

- Um programa que monta um form e aciona outro programa.

```
<?php
printf("<html>\n
        <body>\n
        <form action='p2.php' method='POST'>\n
        Campo N1: <input type='text' name='n1' size=4 maxlength=4>\n
        <input type='submit' value='manda'>\n
        </form>\n
        </body>\n
        </html>\n");
?>
```

- Os desenvolvedores do PHP juram que os dois programas tem a mesma performance... Sinceramente eu acho difícil afirmar.
- A segunda forma talvez apareça um código mais 'limpo', entretanto, rode este programa e no navegador veja o código da página que é recebida pelos dois programas. *Você vai perceber diferença*.

PHP – Estrutura Básica de um PA

- Um programa que monta um form e aciona outro programa.

Código fonte do programa que lê os dados digitados no form:

P2.php

```
<?php
printf("Mostrando o valor: %s<br>\n",$_POST['n1']);
?>
```

O uso dos caracteres de formatação é idêntico à linguagem 'C'

As posições do vetor podem ser indicadas pelos 'nomes' dos campos do form.

O PHP lê os campos de forms em três vetores:

$_POST[], $_GET[] e $_REQUEST[]

PHP – Programas... Estruturas de desvio de fluxo

- Idêntico a linguagem C.
  - if, elseif, else
  - switch
  - for(atribuição;condição;incremento)
  - while
  - do ... While
- Nos programas, o bloco lógico é delimitado por
  "{" – início do bloco
  e
  "}" – fim do bloco
- Vamos mostrar estes comandos mostrando o funcionamento dos programas.

PHP – Programas... Estruturas de desvio de fluxo

- Capturando valores de uma página de formulário:
- Vamos supor uma página com o código:

```
<html>
<head><title>Aprendendo PHP</title></head>
<body>
<form method="post" action="./teste.php"><br>
Desenhador de Tabelas:<br>
<table border=0>
<tr>
  <td>Quantas Linhas</td>
  <td><input type="text" name="LIN" size=10></td></tr>
<tr>
  <td>Quantas Colunas</td>
  <td><input type="text" name="COL" size=10></td></tr>
<tr>
  <td colspan=2><input type="submit" name="sub" value="Enviar!"></td></tr>
</table>
</form>
</body>
</html>
```

Grave estes comandos em um arquivo FORM1.HTML no seu localhost

PHP – Programas... Estruturas de desvio de fluxo

- O código fonte do teste.php pode ser assim:

```
<?php
printf("<html><body>\n");
# Pegando o valor do Vetor $_POST na posição COL
$COL=$_POST['COL'];
$LIN=$_POST['LIN'];
printf("<table border=1>\n");
$i=1;
while ($i<=$LIN)
{
 printf("<tr>");
 for($j=1;$j<=$COL;$j++)
 {
   printf("<td>L=$i, C=$j</td>");
 }
 printf("</tr>\n");
 $i++;
}
printf("</table>\n");
printf("</body></html>\n");
?>
```

Grave este comandos em teste.php e execute http://localhost/form1.html no seu navegador.

PHP – Programas... Estruturas de desvio de fluxo

- Desvios Condicionais
- Criem um arquivo chamado ola.php

Pega a hora da data corrente

Operador lógico: 'E'

Operador elseif igual em shell-Unix ou Perl

```
<?
  $hora = date("H");
  if ($hora >= 0 && $hora < 6)
  {
    printf("<b>Boa Madrugada!!</b>\n");
  }
  elseif ($hora >= 6 && $hora < 12)
  {
    printf("<b>Boa Dia!!</b>\n");
  }
  elseif ($hora >= 12 && $hora < 18)
  {
    printf("<b>Boa Tarde!!</b>\n");
  }
  else
  {
    printf("<b>Boa Noite!!</b>\n");
  }
?>
```

PHP – Programas... Estruturas de desvio de fluxo

- Desvios Condicionais
- Escrevendo o mesmo com o SWITCH

```
<?
  $hora = date("H");
  SWITCH (TRUE)
  {
    case ($hora >= 0 && $hora < 6):
    {
      printf("<b>Boa Madrugada!!</b>\n");break;
    }
    case ($hora >= 6 && $hora < 12):
    {
      printf("<b>Boa Dia!!</b>\n");
      break;
    }
    case ($hora >= 12 && $hora < 18):
    {
      printf("<b>Boa Tarde!!</b>\n");break;
    }
    default:
    {
      printf("<b>Boa Noite!!</b>\n");break;
    }
  }?>
```

O **BREAK** provoca a quebra na execução do bloco de comandos.

Pode ser escrito assim

Ou assim (eu prefiro assim)

PHP – Execução de programas (funções e sub-programas)

- Funções
  - São trechos de programa que serão executados muitas vezes durante a execução de **um** programa
- Sub-Programas
  - São programas que serão executados dentro de um ou mais diferentes programas.
    - Ex. um sub-programa que mostra um cabeçalho de página sempre com alguns dizeres fixos e outros não fixos em uma homepage.
      O programas que vai montar este cabeçalho pode ser 'executado' a partir de diferentes programas que vão montar uma homepage com dados de BD.

PHP – Execução de programas (funções e sub-programas)

- Em PHP um programa pode 'executar' um outro programa de duas formas principais:
  - include
    inclui a execução no mesmo ponto onde o comando está escrito e interrompe a execução do programa principal. Se alguma coisa acontece de errado no subprograma, o programa principal retoma a execução normalmente
  - require
    inclui a execução no mesmo ponto onde o comando está escrito e interrompe a execução do programa principal. Se alguma coisa acontece de errado no subprograma, o programa principal interrompe a execução (é um pouco mais 'seguro' que o include)

PHP – Execução de programas (funções e sub-programas)

- Em PHP um programa pode 'executar' um outro programa de duas formas principais:
  - Se o programa incluído tiver a chamada de uma função e for incluído mais de uma vez no programa principal a referência da mesma função duas vezes vai gerar um conflito de nomes de programas, então o PHP dispõe do ***include_once*** e o ***require_once***. Para 'executar' um programa somente uma vez.

PHP – Execução de programas (funções e sub-programas)

- Montando uma função:
  - Sintaxe

```
function nome_da_função(&par1, par2, par3)
{
    Comandos;
    ... ;
    [return <valor de retorno>];
}
```

- Os valores de pars alterados 'dentro' da função não são percebidos 'fora' da função, a não ser que se use o & na declaração da função, nesta condição o arg1 é lido e seu valor tratado na função é visto no programa principal.

PHP – Execução de programas (funções e sub-programas)

- Exemplificando uma função
-

```
<?
function mais5(&$num1, $num2)
{
 $num1++;
 $num2=8;
 printf("Dentro da Funcao os valores são:<br>\n");
 printf("\$num1: $num1 (\$a+1) e \$num2: $num2<br>\n");
}
$a=$b=1;
printf("Valores ANTES da execução: \$a=$a e \$b=$b<br>\n");
mais5($a, $b); #chamando a função
printf("Valores DEPOIS da execução: \$a=$a e \$b=$b<br>\n");
?>
```

  - /* Neste caso, só $num1 terá seu valor alterado, pois a passagem por referência está definida na declaração da função. */
  - Ao executar este programa...

    Valores ANTES da execução: $a=1 e $b=1
    Dentro da Funcao os valores são:
    $num1: 2 ($a+1) e $num2: 8
    Valores DEPOIS da execução: $a=2 e $b=1

PHP – Execução de programas (funções e sub-programas)

- Funções
  - Sintaxe:com valores pré-definidos (default)

```
function teste($var = “testando”)
{
   printf($var<br>\n");
}
teste();                  // imprime “testando”
teste(“outro teste”); // imprime “outro teste”
```

PHP – Execução de programas (funções e sub-programas)

- Tratando variáveis Globais

Variáveis globais

```
<?
function Teste()
{
 echo $Mostra;
}
$Mostra="teste";
Teste();
?>
```

A função acima não produz saída.
A variável $Mostra é de escopo local e variáveis LOCAIS não podem ser referidas num escopo GLOBAL

Mas... O que é o escopo?

Se quisermos ver o valor da variável dentro da função... Devemos...

```
<?
function Teste()
{
  global $Mostra;
  printf("$Mostra<br>\n");
  $Mostra="Casa";
}
$Mostra="teste";
Teste();
printf("$Mostra<br>\n");
?>
```

Uma boa associação: Escopo é o conceito de processos em SO.
Processo é uma capsula de controle de execução de programas.

PHP – Programa recursivo

- Um programa recursivo é um aquele que 'chama' a ele mesmo em uma referencia de link (que pode ser feita com a tag <a href=' ou com o action na TAG <form).
- Em PHP pode-se usar uma função que verifica se existe valor em uma variável (ou vetor). É a função isset. E depois usar um SWITCH...CASE para escolher os blocos que serão executados.
- O código fica com a estrutura assim:

```
<?php
$bloco=isset($_POST['bloco']) ? $_POST['bloco'] : '1';
SWITCH (TRUE)
{ # 1
  case ($bloco==1):
  { # 1.1 - Monta o form
    break;
  } # 1.1 - Fim do monta form
  case ($bloco==2):
  { # 1.2 - mostra o valor
    break;
  } # 1.2
}
?>
```

PHP – Programa recursivo

P.php

```
<?php
$bloco=(isset($_POST['bloco']) ? $_POST['bloco'] : '1');
SWITCH (TRUE)
{ # 1
  case ($bloco==1):
  { # 1.1 - Monta o form
    printf("<html>\n<body>\n
            <form action='p.php' method='POST'>\n
            <input type='hidden' name='bloco' value='2'>\n
            Campo N1: <input type='text' name='n1' size=4 maxlength=4>\n
            <input type='submit' value='manda'>\n
            </form>\n
            </body>\n</html>\n");
    break;
  } # 1.1 - Fim do monta form
  case ($bloco==2):
  { # 1.2 - mostra o valor
    printf("Mostrando o valor: %s - $_REQUEST[bloco]<br>\n",$_POST['n1']);
    break;
  } # 1.2
}
?>
```

PHP – Variáveis & Constantes

- As variáveis no PHP suportam tipos de dados que podem ser:
  - Inteiro (integer ou long)
  - Ponto flutuante (double ou float)
  - String
  - Array

- Os nomes das variáveis no programa devem se iniciar com o sinal '$'.

PHP – Variáveis & Constantes

- Inteiros (integer ou long)

```
# inteiro positivo na base decimal
$php = 1234;

# inteiro negativo na base decimal
$php = -234;

# inteiro na base octal – iniciado por 0 (=156 decimal)
$php = 0234;

# inteiro na base hexadecimal – iniciado por 0x
$php = 0x34; # 0x34 = 52 decimal.
```

- Ponto flutuante

```
$php = 1.234;
$php = 23e4;     # equivale a 230.000
```

PHP – Variáveis & Constantes

- Strings

```
<?
$teste = "Brasil";
$php = '---$teste--\n';
printf("$php");
?>
```

A saída desse script será:
"---$teste--\n".

```
<?
$teste = "Brasil";
$php = "---$teste---\n";
echo "$php";
?>
```

A saída desse script será:
"---Brasil--"
(com quebra de linha no final).

Coerções (coercividade)

- PHP converte o valor de um deles automaticamente (coerção).

Exemplo:

```
$var = "1";          // $var é do tipo string
$var = $var + 1;     // o operador '+' converte string em inteiro (2)
$var = $var + 3.7;   // agora converteu inteiro para double (5.7)
$var = 1 + 1.5
```

PHP – Variáveis & Constantes

- Constante é um facilitador que a PHP implementa para facilitar o uso de valores fixos dentro de um programa.
- Valem só dentro do programa que está em execução
- Sintaxe:

int **define**(string nome_da_constante, mixed valor);

```
define ("pi", 3.1415926536); // use pi como constante
$circunf = 2*pi*$raio;
```

PHP – Variáveis & Constantes

- Duração de Variável:
  - A variável tem existência durante a execução do programa (dentro do processo que o controla)
  - EXCEÇÃO: Variáveis de ambiente do PHP que perduram entre diferentes programas:
    $GLOBALS
    $_SERVER
    $_GET
    $_POST
    $_FILES
    $_COOKIE
    $_SESSION
    $_REQUEST
    $_ENV
  - Estas são as variáveis 'superglobais'.

PHP – Variáveis & Constantes

- Duração de Variável:
  - E como podemos tornar uma variável 'global'?
    - Escreva: global $NomeVariavel;
  - A variável passa estar disponível para qualquer programa que seja executado 'depois' do programas corrente (na mesma estrutura de processos) ou que tenha executado o programa corrente como função.
    - Se a.php tiver uma global e executar b.php então b.php passa a ver o valor da variável.
    - Se a.php executa b.php como função e em b.php existir uma variável globalizada então a.php passa a ver a variável.

PHP – Os programas ICAEL – montando o frame

- Pode-se pensar em um conjunto de PAs que montam um frame, um menu e executam as ações ICAEL.
- Daqui em diante, mostra-se e analisa-se os PAs que executam estas ações, começando pelo montador de frames.

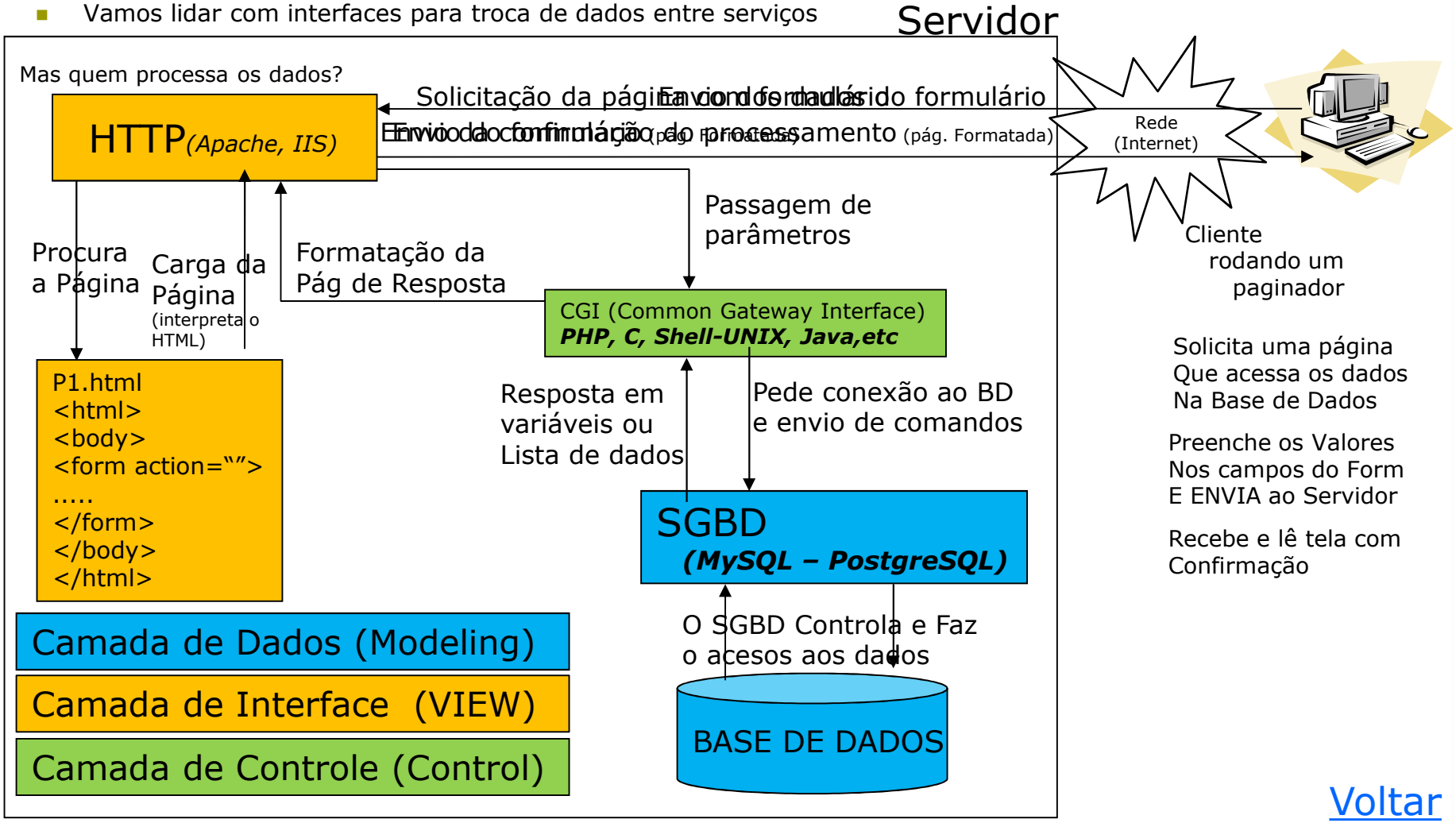
```
<?php
#########################################################################
# Programa....: assisttcphp01.php
# Descricao...: Monta Frames
# Autor.......: JMH
# Objetivo....: Monta um frame horizontal (105,*) pixels, com nomes menu1 e area1.
#               Carrega assisttcphp02.php em menu1 e assisttcphp03.php em area1.
# Criacao.....: 2010-11-10
# Atualizacao.: 2010-11-10
#########################################################################
printf("<html>\n");
printf("<frameset rows='105,*' border=1>\n");
printf("  <frame name='menu1' src='./assisttcphp02.php' noresize scrolling='no'>\n");
printf("  <frame name='area1' src='./assisttcphp03.php' scrolling='auto'>\n");
printf("</frameset>\n");
printf("</html>\n");
?>
```

PHP – Os programas ICAEL – o menu do sistema

- Explicando o menu

```
# Carregando o kit de ferramentas
include("../../fncs/toolskit.php");
$primeira=(isset($_POST['primeira'])? trim($_POST['primeira']):'TRUE');
$cordefundo='navajowhite';
printf("<html>\n<body bgcolor='$cordefundo'>\n");
printf("<font face='tahoma' size=3>\n");
printf("<table  border=0 cellpadding=0 cellspacing=0>\n");
printf("<tr>\n<td valign=top colspan=2>");
printf("<font color=red size=3><b>Equipamentos</b></font></td>\n</tr>\n");
printf("<tr>\n<td valign=top align='right'>\n");
printf("<form action='./equipamephp02.php' method='POST'>\n");
printf("<input type='hidden' name='primeira'  value='FALSE'>\n");
printf("Id: <input type='text' maxlength=4 size=4 name='cpequipamento'>\n");
printf("Nome: <input type='text' maxlength=20  size=20 name='txnomeusual'>\n");
printf("<input type='submit' value='Filtrar'>\n</form>\n</td>\n");
printf("</tr>\n</table>\n");
printf("</font>\n</body>\n</html>\n");
```

- Como se monta um SI em padrão MVC em uma plataforma distribuída?
  - Usamos o conceito de rede de computadores
  - Vamos lidar com interfaces para troca de dados entre serviços



Voltar