

MATERIAL DE APOIO

**AI - APLICATIVOS PARA INTERNET
(PHP COM BANCO DE DADOS)**



Professor: Rafael S. Barreto

SUMÁRIO

INTRODUÇÃO AO PHP	5
COMPARAÇÃO PHP X ASP	5
COMO FUNCIONA?.....	5
ESQUEMA 1 – CÓDIGO HÍBRIDO	5
ESQUEMA 2 – CÓDIGO SEPARADO	6
NOTEBOOK++:	6
WAMP SERVER:.....	6
USANDO O WAMP SERVER:	7
REGRAS PARA SALVAR AS PÁGINAS:	7
EXEMPLO DE CÓDIGO HÍBRIDO:	8
EXEMPLO DE CÓDIGO SEPARADO:	8
COMO ABRIR AS PÁGINAS PARA TESTAR:	9
DIFERENÇAS NOTADAS:	9
PARA EDITAR OS CÓDIGOS PHP:.....	9
COMENTÁRIOS NO PHP.....	10
VARIÁVEIS	11
OPERADORES	12
1. ARITMÉTICOS	12
2. DE STRINGS.....	12
3. DE ATRIBUIÇÃO	12
4. BIT A BIT	13
5. LÓGICOS	13
6. COMPARAÇÃO	13
7. EXPRESSÃO CONDICIONAL	13
8. DE INCREMENTO E DECREMENTO.....	14
FORMAS DE IMPRIMIR TEXTO COM VARIÁVEL	14
FORMULÁRIO DE DADOS.....	15

ATIVIDADES (FORMULÁRIOS COM PHP):	17
ESTRUTURAS DE CONTROLE (DECISÃO E LAÇOS DE REPETIÇÃO).....	18
IF E ELSE	18
IF	18
ELSE.....	18
WHILE, DO WHILE E FOR	19
WHILE	19
DO WHILE	19
FOR	20
BREAK E CONTINUE	20
SWITCH, CASE E DEFAULT.....	21
TRATAMENTO DE FORMULÁRIOS	25
1. CÓDIGO HÍBRIDO - NÃO EXECUTAR O PHP SEM CLIQUE NO BOTÃO.....	25
2. VERIFICAR SE HÁ CAMPOS VAZIOS	26
3. FORMATAR VALORES EM ESTILO MOEDA (R\$).....	26
4. CAIXAS DE MENSAGENS COM JAVASCRIPT	28
5. VALIDAR E-MAIL.....	29
ATIVIDADES (TRATAMENTO DE FORMULÁRIOS):.....	30
FORMULÁRIO COM O NVU.....	31
ATIVIDADES (NVU):.....	41
PHP COM BANCO DE DADOS	42
O PHP MYADMIN	42
USANDO O PHP MYADMIN.....	42
PHP INTERAGINDO COM O SGDB MYSQL.....	45
COMO PHP INTERAGE COM O SGDB MYSQL.....	45
COMO ABRIR UMA CONEXÃO	45
COMO SELECIONAR O BANCO DE DADOS	45
COMO EXECUTAR UMA SQL.....	45
COMO MANIPULAR O RESULTSET DA EXECUÇÃO ANTERIOR	46
COMO FECHAR A CONEXÃO	46
MANIPULAÇÃO DO BANCO DE DADOS.....	46
CADASTRAR (GRAVAÇÃO)	47
LISTAR TODOS OS DADOS (CONSULTA GERAL)	49
CONSULTA (BUSCA)	50

EXCLUSÃO (APAGAR REGISTRO).....	52
SORTEAR INFORMAÇÕES DO BANCO DE DADOS	54
LOGIN COM SESSÃO	56
CONEXÃO:.....	57
CADASTRO:	58
AUTENTICAÇÃO:	59
PÁGINA RESTRITA:.....	60
VERIFICAÇÃO DE AUTENTICAÇÃO:	60
EFETUAR LOGOFF:	60
ATIVIDADES (PHP COM BANCO DE DADOS):.....	60
EXPORTAÇÃO E IMPORTAÇÃO DE BANCO DE DADOS NO PHP MYADMIN	61
EXPORTANDO O BANCO DE DADOS PRONTO:	61
IMPORTANDO O BANCO DE DADOS EM OUTRA MÁQUINA:	63

INTRODUÇÃO AO PHP

Ao contrário do que muita gente pensa, **HTML** (Hyper Text Markup Language) não é uma linguagem de programação e sim uma linguagem de escrita e formatação de hipertexto.

Para ser considerada uma linguagem de programação, é preciso ter suporte a variáveis, tomadas de decisão, laço de repetição, ou ter acesso a Banco de Dados e isto o HTML não permite. Mas com a popularização e a evolução da Internet surgiu a necessidade de se criar sites dinâmicos que manipulassem informações.

O PHP, assim como o ASP, surgiu para solucionar este problema e dar às páginas de Internet as funções que elas não tinham.

PHP significa – **P**H_P **H**ypertext **P**rocessor ou **P**ersonal **H**ome **P**age. É uma linguagem interpretada, ou seja, roda em tempo real, sem a necessidade de ser transformada em linguagem de máquina (compilação).

COMPARAÇÃO PHP X ASP

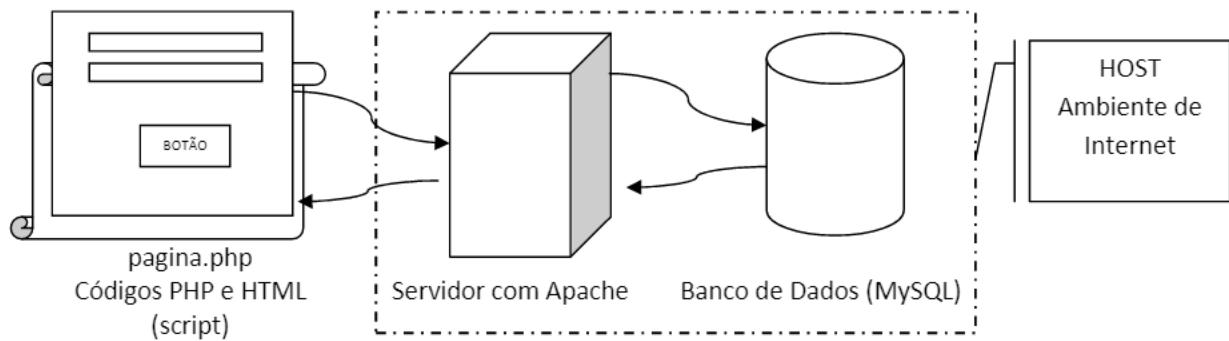
ASP	PHP
<ul style="list-style-type: none"> • Servidor Windows • Maior custo • Mais complicada 	<ul style="list-style-type: none"> • Servidor Windows ou Linux • Menor custo • Mais fácil

COMO FUNCIONA?

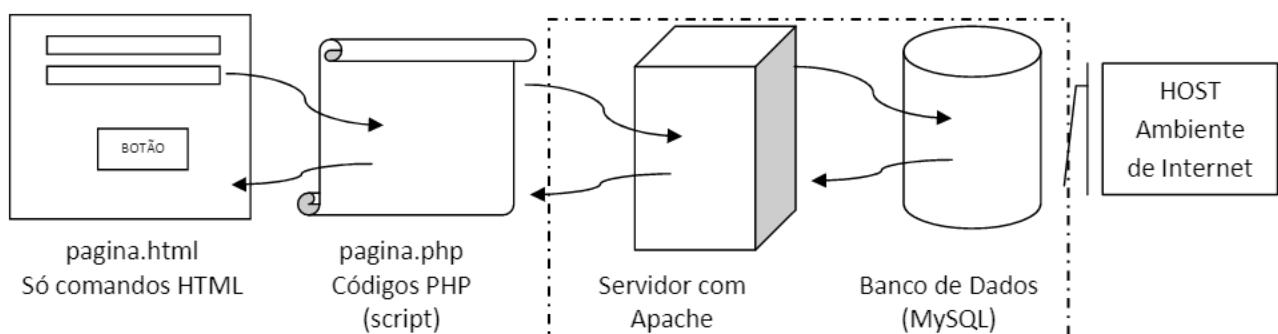
O browser (navegador Internet Explorer, Firefox, Chrome, etc) lê os códigos HTML e envia os dados para um servidor Apache, que interpreta os comandos PHP e devolve a resposta ao navegador. Se houver a necessidade de trabalhar com Banco de Dados (MySQL é o mais usado), o Apache também cuida desta ação.

As páginas podem ser feitas em código híbrido (HTML e PHP em um único arquivo) e código separado (um HTML ligado a um PHP). Inicialmente abordaremos os dois modelos para todos os exemplos e, posteriormente utilizaremos conforme a necessidade e facilidade de manipulação.

ESQUEMA 1 – CÓDIGO HÍBRIDO



ESQUEMA 2 – CÓDIGO SEPARADO



NOTE PAD++:

Para criar nossos códigos, vamos utilizar um bloco de notas com recursos avançados para diversas linguagens de programação. A vantagem é que ele numera as linhas de código, o que ajuda a achar os erros de PHP e ainda reconhece comandos e variáveis, podendo até mesmo invocar comandos.

O Notepad++ pode ser baixado em www.superdownloads.com.br.

A instalação é muito simples e a da primeira vez em que for aberto, ele exibirá um relatório que poderá ser fechado.

WAMP SERVER:

www.profralph.com.br

Simula um servidor de Internet com Apache e MySQL na sua máquina, para que você teste as páginas sem hospedar.

Existem outros programas que fazem a mesma coisa, como o PHPTriad, EasyPHP, Xampp, etc. Utilizaremos o Wamp pela facilidade de uso.

1º passo - Você deve fazer download do software em:

<http://www.wampserver.com/dl.php> ou www.superdownloads.com.br

2º passo – execute o arquivo que você baixou, ele vai perguntar se você deseja continuar a instalação, clique em SIM.

3º passo – será exibido a tela de apresentação, clique em NEXT, logo após leia o Termos de Uso e clique novamente em NEXT.

4º passo – escolha onde será instalado o software, é altamente recomendável deixar o c:\wamp como padrão, pois nomes com espaços ou maiores que 8 dígitos poderão ser problemas no futuro, na hora de configuração dos servidores. ahh.. clique em NEXT, duas vezes.

5º passo – Adivinha? Sim, aperte INSTALL.

6º passo – Você deve informar o programa, qual o seu navegador padrão. Se você deseja utilizar o Internet Explorer 7, você deve ir no caminho C:\Arquivos de programas\Internet Explorer e escolher o arquivo iexplore.exe, mas caso você seja mais esperto e já possui o Firefox em seu computador, escolha o caminho C:\Arquivos de programas\Mozilla Firefox e selecione o arquivo firefox.exe ... depois de escolher uma dessas opções, clique em OK.

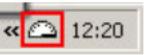
7º passo - Você deve informar o servidor SMTP (servidor de e-mail), mas como estamos instalando para fins educativos inicialmente deixaremos esta opção como padrão, apenas clique em NEXT.

8º passo – Instalação concluída! Clique em FINISH para executar o WampServer.

9º passo – O WS esta rodando em seu sistema, para utiliza-lo você deverá encontrar um ícone ao lado do seu relógio (do Windows), clique sobre o botão direito, vá até LANGUAGE e selecione PORTUGUES.

10º passo – Agora clique uma vez com o botão esquerdo sobre o mesmo ícone e verá todas as opções do programa, como parar, reiniciar e iniciar os serviços (php, mysql, apache).

USANDO O WAMP SERVER:

Clicar com o botão esquerdo do mouse no ícone  na Área de Notificação (ao lado do relógio do Windows) para abrir as opções, das quais usaremos:

- **LOCALHOST** – Para acessar e testar as páginas
- **PHPMYADMIN** – Para criar e acessar Banco de Dados
- **WWW DIRECTORY** – Onde as páginas são gravadas
- **STOP ALL SERVICES** – Interrompe os serviços do Apache e MySQL
- **START ALL SERVICES** – Conectar com o Apache e MySQL
- **RESTART ALL SERVICES** – Reiniciar os serviços

OBS.: Quando o desenho do Wamp estiver branco, tudo está funcionando corretamente, se estiver amarelo, ou o Apache, ou o MySQL estão com problemas e quando estiver vermelho, nada está funcionando. Para tentar solucionar, deve-se parar os serviços e iniciar novamente ou fechar e reabrir o Wamp. Se ainda assim não funcionar, deve-se tentar desinstalar o Wamp e instalar outra versão.

REGRAS PARA SALVAR AS PÁGINAS:

- **SEMPRE** salvar numa pasta criada em www (no Wamp ou Xampp) ou htdocs (no EasyPHP ou PHPTriad)
- **CADA PROJETO** na sua pasta específica
- **NUNCA** usar letras maiúsculas, espaços ou símbolos especiais nos nomes das pastas, páginas ou scripts
- **PARA CÓDIGO HÍBRIDO**, salvar com a extensão .php
- **PARA CÓDIGO SEPARADO**, salvar as páginas como .html e a programação como .php

EXEMPLO DE CÓDIGO HÍBRIDO:

Digitar no Notepad ++ e salvar na www/exemplo_hibrido com o nome **data_hoje.php**

```
<html>
<head><title>Exemplo 1 de PHP-Híbrido</title></head>
<body>
<?php
    $data_de_hoje=date("d/m/Y",time());
?>
<p align="center">
Hoje é dia <?php echo "$data_de_hoje"; ?>
</p>
</body></html>
```

www.profralph.com.br

EXEMPLO DE CÓDIGO SEPARADO:

CÓDIGO HTML: Digitar no Notepad ++ e salvar na www/exemplo_separado com o nome **data_hoje.html**

```
<html>
<head><title>Exemplo 1 de PHP-Separado</title></head>
<body>
<p align="center">
Clique <a href="data_hoje.php"> aqui </a> para ver a data de hoje.
</p>
</body></html>
```

CÓDIGO PHP: Digitar no Notepad ++ e salvar na www/exemplo_separado com o nome **data_hoje.php**

```
<?php
    $data_de_hoje=date("d/m/Y",time());
    echo "Hoje é dia $data_de_hoje";
?>
```

ATENÇÃO! Perceba que nos dois exemplos temos comandos entre **<?php** e **?>**. Isto indica onde começa e onde termina o PHP.

No PHP variáveis não precisam ser declaradas, basta começar o nome com \$, como no caso da variável **\$data_de_hoje**.

date("d/m/Y",time()); é uma função que pega a data do sistema.

Echo é um comando para escrever no navegador, portanto **echo "Hoje é dia \$data_de_hoje";** mostraria a data que foi armazenada pela função na variável com o comando

\$data_de_hoje=date("d/m/Y",time());

Outro detalhe importante é o uso do ponto e vírgula no final dos comandos PHP.

COMO ABRIR AS PÁGINAS PARA TESTAR:

As páginas em PHP não podem ser abertas de qualquer lugar e simplesmente clicando duas vezes. Elas precisam ser interpretadas, ou seja, você é obrigado a abrir como se estivesse navegando na Internet. Antes de abri-las, não se esqueça de abrir o Wamp. **Quando você testar uma página em PHP e aparecer uma tela de download, para salvar o arquivo PHP, significa que você não está abrindo corretamente.**



Para abrir da maneira correta, clique no ícone e escolha a opção Localhost.

ATENÇÃO! Em alguns casos, ao invés de abrir a página do Wamp, seu navegador pode abrir um site de endereço www.localhost.com.br. Se este problema acontecer, vá à barra de endereços do seu navegador e substitua este endereço por **//127.0.0.1**, assim, será carregada a página do Wamp pelo ip da sua placa de rede e você poderá trabalhar normalmente.

Na página inicial do Wamp, procure pelo link Your Projects (Seus Projetos) e clique no link do projeto que você irá testar.

DIFERENÇAS NOTADAS:

No código híbrido tudo é executado na mesma página e em tempo real, muitas vezes sem a necessidade de intervenção do usuário, a não ser em formulários, enquanto no código separado o HTML vai para outra página PHP para enfim executar os comandos.

Apesar deste detalhe, não há diferença em velocidade de execução dos comandos.

PARA EDITAR OS CÓDIGOS PHP:

Não é possível utilizar o comando Exibir, Código fonte do navegador, pois o PHP desta forma não mostra os comandos e sim os resultados dos comandos, por exemplo, no código **data_hoje.php**, exibiria: **Hoje é dia xx/xx/yyyy**

Onde x seriam os valores de dia, mês e ano mostrados.

Para fazer alteração no código, devemos ir até o diretório www, clicando no ícone do wamp e abrindo a opção **www directory**. A partir daí, abrir a pasta onde está o projeto, clicar com o botão direito do mouse e escolher a opção Edit with no Notepad ++.

Altere o código como no exemplo a seguir:

```
<?php
    echo "<center>";
    echo "Olá visitante!";
    $data_de_hoje=date("d/m/Y",time());
    echo "Hoje é dia <font color='red'>$data_de_hoje</font>";
?>
```

Salve (CTRL+S) e abra novamente para ver o resultado.

ATENÇÃO! PHP não conhece comandos de HTML, por isso foi necessário colocar os comandos entre o **echo** “ ”;. O echo engana o PHP, pois quando ele vai mandar os resultados, o navegador entende que o PHP está escrevendo os comandos HTML.

COMENTÁRIOS NO PHP

Podemos fazer comentários simples (uma linha) ou em bloco (várias linhas). **Mas lembre-se!** Os comentários mostrados a seguir só funcionam dentro do PHP, ou seja, entre **<?php** e **?>**. Colocando os comentários fora do PHP fará com que seja mostrado como texto comum.

Para comentário simples utilizamos o comando **//**, como no exemplo:

```
<?php
    echo "<center>";
    echo "Olá visitante!"; //SAUDAR O VISITANTE
    //PEGAR A DATA E JOGAR NA VARIÁVEL $data_de_hoje
    $data_de_hoje=date("d/m/Y",time());
    //MOSTRAR A DATA FORMATADA
    echo "Hoje é dia <font color='red'>$data_de_hoje</font>";
?>
```

Para comentários em blocos utilizamos **/*** no começo e ***/** no fim, como no exemplo:

```
<?php
/*echo "<center>";
echo "Olá visitante!";*/
//COMENTEI ACIMA PORQUE NÃO QUERO MAIS CENTRALIZAR E SAUDAR O VISITANTE
$data_de_hoje=date("d/m/Y",time());
echo "Hoje é dia <font color='red'>$data_de_hoje</font>";
?>
```

VARIÁVEIS

As variáveis no PHP não exigem declaração e aceitam qualquer tipo (conteúdo) em qualquer momento. Devemos apenas nos lembrar de dois detalhes cruciais:

Variáveis iniciam **SEMPRE** com o caractere \$.

PHP é key sensitive, ou seja, diferencia letras maiúsculas de minúsculas, portanto, se sua variável tiver letras minúsculas, deve se usar assim sempre, o mesmo com letras maiúsculas. Para o PHP, as variáveis **\$nome**, **\$Nome**, **\$NOme**, **\$NOMe** e **\$NOME** são cinco variáveis diferentes.

Exemplo1: Digitar no Notepad ++ e salvar na www/exemplos com o nome variaveis.php

```
<?php
```

```
$n1=10+7;  
echo "O valor é: $n1";  
echo "<br>";  
$n1= "Rafael";  
Echo "O nome é: $n1";
```

```
?>
```

Imprime no navegador primeiro a frase “O valor é: 17” que está dentro de \$n1 e em seguida, \$n1 passa a valer o nome (Rafael) e seria impresso “O nome é: Rafael”. Podemos reparar então que a variável consegue assumir valor de número e texto.

Exemplo2: Digitar no Notepad ++ e salvar na www/exemplos com o nome erro_variaveis.php

```
<?php
```

```
$n1=10;  
echo "O valor é: $N1";
```

```
?>
```

Imprimaria no navegador apenas **O valor é:**, pois \$n1 vale 10, enquanto \$N1 estará vazia.

OPERADORES

Operadores são caracteres especiais digitados para criarmos expressões ou trabalharmos com testes lógicos. Os mais importantes são:

1. ARITMÉTICOS

Só podem ser utilizados quando os operandos são números (integer ou float). Se forem de outro tipo, terão seus valores convertidos antes da realização da operação:

- + - adição;
- - subtração;
- * - multiplicação;
- / - divisão;
- % - módulo.

2. DE STRINGS

Só há um operador exclusivo para strings:

- . - concatenação.

3. DE ATRIBUIÇÃO

Existe um operador básico de atribuição e diversos derivados. Sempre retornam o valor atribuído. No caso dos operadores derivados de atribuição, a operação é feita entre os dois operandos, sendo atribuído o resultado para o primeiro. A atribuição é sempre por valor, e não por referência:

- = - atribuição simples;
- += - atribuição com adição;
- = - atribuição com subtração;
- *= - atribuição com multiplicação;
- /= - atribuição com divisão;
- %= - atribuição com módulo;
- .= - atribuição com concatenação.

Exemplo:

```
$a = 7;  
$a += 2; // $a passa a conter o valor 9
```

4. BIT A BIT

Comparam dois números bit a bit:

& - "e" lógico;
| - "ou" lógico;
^ - ou exclusivo;
~ - não (inversão);
<< - shift left;
>> - shift right.

5. LÓGICOS

Utilizados para inteiros representando valores booleanos:

and - "e" lógico;
or - "ou" lógico;
xor - ou exclusivo;
! - não (inversão);
&& - "e" lógico;
|| - "ou" lógico.

Existem dois operadores para "e" e para "ou" porque eles têm diferentes posições na ordem de precedência.

6. COMPARAÇÃO

As comparações são feitas entre os valores contidos nas variáveis, e não as referências. Sempre retornam um valor booleano:

== - igual a;
!= - diferente de;
< - menor que;
> - maior que;
<= - menor ou igual a;
>= - maior ou igual a.

7. EXPRESSÃO CONDICIONAL

Existe um operador de seleção que é ternário. Funciona assim:

(expressao1)?(expressao2):(expressao3)

O interpretador PHP avalia a primeira expressão. Se ela for verdadeira, a expressão retorna o valor de expressão2. Senão, retorna o valor de expressão3.

8. DE INCREMENTO E DECREMENTO

Operadores de incremento e decremento:

++ - incremento;
-- - decremento.

Podem ser utilizados de duas formas: antes ou depois da variável. Quando utilizado antes, retorna o valor da variável antes de incrementá-la ou decrementá-la. Quando utilizado depois, retorna o valor da variável já incrementado ou decrementado.

Exemplos:

```
$a = $b = 10; // $A E $B RECEBEM O VALOR 10  
$c = $a++; // $C RECEBE 10 E $A PASSA A TER 11  
$d = ++$b; // $D RECEBE 11, VALOR DE $B JÁ INCREMENTADO
```

FORMAS DE IMPRIMIR TEXTO COM VARIÁVEL

O PHP consegue diferenciar o que é texto do que é variável, desde que para a impressão seja feito o uso de aspas "".

O apóstrofo serve para impressão literal, ou seja, acontece a impressão literal, ou seja, a variável vai aparecer como um texto normal.

Caso haja confusão entre o uso de ambos, podemos ainda trabalhar com concatenação, ou seja, junção de texto com valor de variável.

Exemplo: Digitar no Notepad ++ e salvar na www/exemplos com o nome mostrar.php

<?php

```
$n1=10+7;  
echo 'O valor é: $n1';  
echo "<br>";  
$n1= "Rafael";  
echo "O nome é: $n1";  
echo "<br>";  
$n1=date("d/m/Y",time());  
echo 'A data é:'.$n1;
```

?>

Ao testarmos, podemos perceber que o resultado é:

O valor é: \$n1

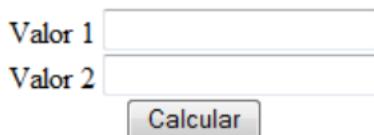
O nome é: Rafael

A data é: xx/xx/yyyy

Vemos que no primeiro caso ocorre uma impressão literal por causa do apóstrofo que não difere texto de variável. Nos segundo caso há o reconhecimento de texto e variável devido ao uso de aspas e finalmente no terceiro caso usamos apóstrofo, mas concatenamos (com .) a expressão com a variável, que está fora dos apóstrofos.

FORMULÁRIO DE DADOS

Formulários são telas feitas no HTML com campos para passar valores para o PHP, como a figura a seguir:



Valor 1
Valor 2
Calcular

Trabalhar com formulários exige atenção a alguns detalhes **MUITO IMPORTANTES**:

- **DECLARE SEMPRE** seus formulários antes de colocar botões, caixas de texto, etc, para que os botões possam funcionar e o PHP consiga receber os dados;
- **A AÇÃO** é sempre o arquivo .php se o código for híbrido é o mesmo nome do arquivo;
- **INFORME** o método de envio de dados;

ATENÇÃO! Existem dois métodos de envio de dados:

GET – método padrão. Os dados são enviados para a barra de endereço do navegador e depois para o PHP.

POST – método protegido. A barra de endereço não recebe os dados, assim não ficam visíveis aos usuários.

Se você não declarar nenhum método o PHP assumirá o GET, isto significa que as informações como senha, por exemplo, ficarão visíveis.

- **DÊ NOMES** a todos os objetos do formulário;
- **BOTÕES DE RÁDIO** (bolinhas, ou options) tem sempre o mesmo nome, não importa quantos apareçam para uma mesma opção;
- **LEMBRE-SE** que o PHP é key sensitive, os nomes forem dados com letras minúsculas, deve se usar assim sempre, o mesmo com letras maiúsculas.

Faremos um formulário simples com código híbrido e separado, apenas para ver como é o processo.

Exemplo1 – CÓDIGO SEPARADO:

CÓDIGO HTML: Digitar no Notepad ++ e salvar na www/exemplos/separado com o nome **form_calcula.html**

```
<html>
<head><title>Exemplo Form-Separado</title></head>
<body>
<center>
<form action='form_calcula.php' method='post'>
Valor 1 <input type='text' name='txt_v1'><br>
Valor 2 <input type='text' name='txt_v2'><br>
<input type='submit' name='btn_calc' value='Calcular'><br>
</center>
</body>
</html>
```

OBS: Perceba que o form action é o mesmo nome do nosso código a seguir, ou seja, **form_calcula.php** e é isto que faz o PHP conseguir “conversar” com o HTML quando o código é separado. Pode ser usado qualquer nome, desde que se respeitem as regras (minúsculo, sem espaços, etc) e que seja o mesmo dado ao PHP e colocado no action do HTML.

CÓDIGO PHP: Digitar no Notepad ++ e salvar na www/exemplos/separado com o nome **form_calcula.php**

CUIDADO! O comando **\$_POST** obrigatoriamente deve ser escrito em **maiúsculo**, pois só funciona desta maneira.

LEMBRE-SE que se no lugar de method='POST' no HTML estiver method='GET', no PHP este comando será **\$_GET**. Não dá para declarar um método **POST** no HTML e receber com **GET** no PHP e vice versa. Cometer este erro fará com que o PHP não consiga receber os dados e as variáveis ficarão vazias.

```
<?php
$v1=$_POST['txt_v1']; //JOGANDO A CAIXA txt_v1 NA VARIÁVEL $v1
$v2=$_POST['txt_v2']; //JOGANDO A CAIXA txt_v2 NA VARIÁVEL $v2
$total=$v1+$v2; //CALCULANDO A SOMA
echo "A soma dos dois valores é:$total"; //MOSTRANDO O RESULTADO
?>
```

Exemplo2 – CÓDIGO HÍBRIDO: Digitar no Notepad ++ e salvar na www/exemplos/hibrido com o nome **form_calcula.php**

```
<html>
<head><title>Exemplo Form-Híbrido</title></head>
<body>
<center>
<form action='form_calcula.php' method='post'> //ACTION É O NOME DO ARQUIVO PHP
Valor 1
<input type='text' name='txt_v1'><br>
Valor 2
<input type='text' name='txt_v2'><br>
<input type='submit' name='btn_calc' value='Calcular'><br>
<?php
$V1=$_POST['txt_v1'];
$V2=$_POST['txt_v2'];
$Total=$V1+$V2;
echo "A soma dos dois valores é:$Total";
?>
</center>
</body>
</html>
```

OBS: Perceba que mesmo sendo um código híbrido o form action é obrigatoriamente usado e que é o mesmo nome do nosso arquivo, ou seja, **form_calcula.php** (**estamos ligando o arquivo a ele mesmo**) e é isto que faz o PHP conseguir “conversar” com o HTML aqui também. Pode ser usado qualquer nome, desde que se respeitem as regras (minúsculo, sem espaços, etc) e que seja o mesmo dado ao PHP e colocado no action do HTML.

ATIVIDADES (FORMULÁRIOS COM PHP):

1. Criar um formulário para receber o nome do funcionário, o valor do salário hora e a quantidade de horas trabalhadas. O PHP deverá calcular o salário bruto, o INSS (11%) e o salário líquido. Todas as informações deverão ser impressas.
2. Criar um formulário para receber o nome do pagador, o valor da parcela, data de vencimento e dias em atraso. O PHP deverá calcular juros por dia (sendo 13% do valor) e o total a pagar. Todas as informações deverão ser impressas.

OBS.: Fazer para ambos duas versões, uma com código híbrido e uma com código separado. Para calcular os percentuais não use 11% e 13% na operação e sim 0.11 e 0.13.

ESTRUTURAS DE CONTROLE (DECISÃO E LAÇOS DE REPETIÇÃO)

IF E ELSE

Um fator de extrema importância em qualquer linguagem de programação é a capacidade de tornar nosso código capaz de tomar decisões baseando-se no que é falso e verdadeiro (lógica booleana).

Em PHP, usamos if e else para tal ação, em conjunto com operadores lógicos para que possamos fazer comparações.

IF

If verifica se uma comparação ou sentença é verdadeira, se for, ele executará a próxima linha ou o próximo bloco de dados dentro de chaves ({}).

Para tal comparação, usamos os operadores lógicos:

- > : Maior
- < : Menor
- >= : Maior ou igual
- <= : Menor ou igual
- == : Igual
- != : Diferente

ELSE

Else apenas será executado caso a comparação ou sentença do if for falso. Portanto, else só existe se existir um if primeiro.

Else não é uma cláusula obrigatória, portanto, pode existir if sem ter necessariamente um else.

Abaixo está um exemplo simples de como usar if e else:

```
<?php
/* CÓDIGO QUE VERIFICA SE UM NÚMERO É PAR OU ÍMPAR */
$numero = 50;
if ($numero%2 == 0)
/*
* SE O RESTO DA DIVISÃO DE UM NÚMERO POR 2 FOR 0,
* ENTÃO, QUER DIZER QUE ESTE NÚMERO É PAR.
```

```
**/  
echo ("O número $numero é par");  
else  
/* CASO CONTRÁRIO */  
echo ("O número $numero é ímpar");  
?>
```

WHILE, DO WHILE E FOR

While, Do While e For são comandos que servem para manter uma porção do código executando repetidamente.

Também conhecidos como estruturas de repetição, iterações ou loops, esses comandos mantém a execução até que seu argumento seja falso.

WHILE

A estrutura principal de while é: while (comparação) { código que será executado }. Ou seja, a comparação é verificada, caso ela seja verdadeira, então, o código é executado.

No exemplo abaixo, faremos um código para contar de 1 a 10.

```
<?php  
$nr = 1;  
while ($nr<=10) {  
    echo ($nr . " ");  
    $nr++;  
}  
?>
```

Isso irá resultar em: 1 2 3 4 5 6 7 8 9 10

DO WHILE

Do while funciona da mesma forma que while, mas ambos têm uma pequena diferença. Como foi visto, o comando while verifica a variável primeiro, mas do while faz o contrário, ele executa o código primeiro e só depois verifica a variável, se ela for verdadeira, então ele executa o código novamente.

A estrutura principal de do while é: do { código que será executado } while (comparação);

Então, se fossemos reescrever o código acima usando do while ficaria:

```
<?php  
$nr = 1;  
do {  
    echo ($nr . " ");  
    $nr++;  
} while ($nr<=10)  
?>
```

Precauções: Como podemos observar tanto while quanto do while fazem com que o código fique repetindo o mesmo trecho, portanto, temos que nos assegurar que a comparação ou variável que está em while de alguma forma se torne falsa. Isso porque se a variável for sempre verdadeira, a execução nunca irá terminar, ocorrendo o que chamamos de looping infinito. Felizmente, há parâmetros na configuração do PHP que limitam o tempo de execução do código que, por padrão, são 30 segundos.

FOR

A estrutura for também cria iterações (repetições) e tem o mesmo problema do while, mas diminui o risco de looping infinito porque for controla toda a variável, desde seu valor inicial até seu valor final.

A estrutura principal de for é: for (valor inicial; comparação; modificação do valor) { código que será executado }

Ou seja, diferentemente de while e do while que tínhamos que declarar a variável já contendo um valor (`$nr = 1;`) e dentro do bloco de execução incrementar esta variável (`$nr++;`), o for já faz tudo isso para nós.

```
<?php  
for ($nr=1; $nr<=10; $nr++) {  
    echo ($nr . " ");  
}  
?>
```

BREAK E CONTINUE

Break e continue são utilizados para controlar estruturas como while, do while, for e switch.

A função de break é forçar a saída de uma estrutura. Por exemplo, se tivermos uma estrutura de repetição for que vai até um determinado número, mas que deve parar antes caso alguma condição seja satisfeita, neste caso usamos o break. Veja o exemplo:

```
<?php  
  
for ($i=2; $i <= 100; $i+=2) { //números pares de 2 a 100  
    echo ($i . " ");  
  
    if ($i%10 == 0) //SE I FOR DIVISÍVEL POR 10  
        break;      //SAIA DO LOOP  
  
}  
  
?>
```

Resultado: 2 4 6 8 10

O código acima mostra exatamente a funcionalidade do comando `break`.

Nós definimos a estrutura `for` para que mostrasse os números pares de 2 a 100, mas apenas serão apresentados os pares de 0 a 10. Isso acontece porque há uma condição dentro da estrutura que diz que se a variável `i` for divisível por 10, então, executa o comando `break`.

No momento que `break` é executado, não estamos mais dentro do loop.

O comando `continue` funciona de forma parecida com o comando `break`, mas ao invés de sair completamente da estrutura, ele apenas ignora o resto do código. Vejamos:

```
<?php  
  
for ($i=0; $i <= 10; $i++) {  
  
    if ($i%2 == 0) //SE I FOR PAR (DIVISÍVEL POR 2)  
        continue; //IGNORA O RESTO DO FOR E VAI PARA O PRÓXIMO NÚMERO  
  
    echo ($i . " ");  
  
}  
  
?>
```

O que resulta em: 1 3 5 7 9

Ou seja, toda vez que `i` for divisível por 2, então, ignorar o resto do código (`echo ($i . " ");`) como se já tivesse chegado na chave final do `for`.

SWITCH, CASE E DEFAULT

Em PHP, existe um comando que permite utilizar várias opções definidas agrupadas em uma única estrutura. Estamos falando de `switch`.

`Switch` trabalha usando casos (`case`) que são valores definidos.

A estrutura do `switch` é: `switch (variável) { case valor: código a ser executado }`

O importante é entender que o switch executa tudo a partir do ponto do case encontrado, isso quer dizer que, se possuirmos um switch com 5 cases e o segundo case for ativado, então, será executado o case 2 e também os cases 3, 4 e 5. Para evitar que isso aconteça, podemos simplesmente parar a execução do switch usando o comando break.

Switch nos poupa o trabalho de criar várias estruturas if-else aninhadas se elas forem apenas para verificar um dado específico. Por exemplo, se quizéssemos que o usuário digitasse um número entre 0 a 4 apenas para executar um trecho de código.

```
<?php  
  
$var1 = 0;  
  
if ($var1 == 0) {  
  
    //CÓDIGO SE VARI FOR 0  
  
} elseif ($var1 == 1) {  
  
    //CÓDIGO SE VARI FOR 1  
}  
elseif ($var1 == 2){  
  
    //CÓDIGO SE VARI FOR 2  
  
} elseif ($var1 == 3) {  
  
    //CÓDIGO SE VARI FOR 3  
  
} elseif ($var1 == 4) {  
  
    //CÓDIGO SE VARI FOR 4  
  
}  
  
?>
```

Com switch seria assim:

```
<?php  
  
$var1 = 0;  
  
switch ($var1) {  
  
    case 0:  
  
        //CÓDIGO SE VARI FOR 0  
  
        break;  
  
    case 1:  
  
        //CÓDIGO SE VARI FOR 1  
  
        break;  
  
    case 2:
```

```
//CÓDIGO SE VAR1 FOR 2
break;
case 3:
//CÓDIGO SE VAR1 FOR 3
break;
case 4:
//CÓDIGO SE VAR1 FOR 4
break;
}
?>
```

O caso default serve para executar um trecho do código que não tenha seu valor definido em case.

Usando o exemplo acima, o switch possui 5 cases que vão de 0 a 4, o que aconteceria se a variável var1 valesse 10? Está é a finalidade do default, ele irá pegar qualquer caso que não esteja definido.

```
<?php
$var1 = 0;
switch ($var1) {
    case 0:
        //CÓDIGO SE VAR1 FOR 0
        break;
    case 1:
        //CÓDIGO SE VAR1 FOR 1
        break;
    case 2:
        //CÓDIGO SE VAR1 FOR 2
        break;
    case 3:
        //CÓDIGO SE VAR1 FOR 3
        break;
    case 4:
```

```
//CÓDIGO SE VAR1 FOR 4
break;

default:
//CÓDIGO SE VAR1 NÃO FOR NENHUM DOS CASOS ANTERIORES
break;
}

?>
```

Abaixo está um exemplo de como podemos usar switch, case e default de forma simples.

```
<?php
$letra = 'A';
switch (strtoupper($letra)) {
    case 'A':
    case 'E':
    case 'I':
    case 'O':
    case 'U':
        echo ("Esta letra é uma vogal");
        break;
    default:
        echo ("Esta letra é uma consoante");
        break;
}
```

?>

No exemplo acima, há um switch que irá procurar pelo valor da variável *letra*. Como podemos ver, *letra* vale A. Como PHP é uma linguagem que chamamos de case sensitive (diferencia maiúsculas de minúsculas) usamos a função strtoupper() para converter qualquer caracter em maiúsculo. Switch irá encontrar logo no primeiro caso o valor A e começará a executar dessa linha em diante, pois não há nenhum comando que pare a execução, ou seja, o switch irá executar até o case U onde mostrará uma mensagem -> Esta letra é uma vogal. Mas não será executado default porque há um comando break antes dele que impedirá sua execução. No entanto, se modificássemos o valor de *letra* para qualquer caracter diferente de A, E, I, O e U, apenas default seria executado.

Fonte: <http://www.tiexpert.net/programacao/web/php/index.php>

TRATAMENTO DE FORMULÁRIOS

1. CÓDIGO HÍBRIDO - NÃO EXECUTAR O PHP SEM CLIQUE NO BOTÃO

Quando trabalhamos com código híbrido e no código PHP há instruções para impressão de mensagens ou cálculos estes poderão ser mostrados logo que a página é aberta.

Como vimos no Exemplo 2 anterior, é impressa na tela “A soma dos dois valores é: 0”. Isto porque a página foi carregada com campos vazios, portanto a soma de campos vazios é 0. Mas podemos impedir o PHP de já iniciar a impressão de mensagens ou resultados quando os campos estiverem vazios e só mostrá-los quando preenchermos os campos. Para tanto, basta fazer algumas alterações no código, como mostrado a seguir no nosso arquivo **form_calcula.php**:

```
<html>
<head><title>Exemplo Form-Híbrido</title></head>
<body>
<center>
<form action='form_calcula.php' method='post'>
    Valor 1
    <input type='text' name='txt_v1'><br>
    Valor 2
    <input type='text' name='txt_v2'><br>
    <input type='submit' name='btn_calc' value='Calcular'><br>
<?php
    $v1=$_POST['txt_v1'];
    $v2=$_POST['txt_v2'];
    $BTN=$_POST['btn_calc']; //JOGANDO O BOTÃO NA VARIÁVEL $BTN (ESTE É O SEGREDO)
    if ($BTN!=''){
        $total=$v1+$v2; //SOMAR
        echo "A soma dos dois valores é:$total"; //MOSTRAR
    } //FECHA A CHAVE ABERTA NO IF
?>
</center>
</body>
</html>
```

OBS: Perceba que os comentários (com //) são usados apenas entre <?php e ?>.

2. VERIFICAR SE HÁ CAMPOS VAZIOS

Podemos fazer com que o PHP verifique se todos os campos foram preenchidos. Esta ação pode poupar trabalho posteriormente, quando ao trabalharmos com banco de dados quisermos exigir que todos os campos sejam preenchidos pelo visitante do site.

Este tratamento serve tanto para código híbrido quanto para código separado.

Faremos agora outras alterações no código, como mostrado a seguir no nosso arquivo **form_calcula.php**:

```
<html>
<head><title>Exemplo Form-Híbrido</title></head>
<body><center>
<form action='form_calcula.php' method='post'>
Valor 1
<input type='text' name='txt_v1'><br>
Valor 2
<input type='text' name='txt_v2'><br>
<input type='submit' name='btn_calc' value='Calcular'><br>
<?php
$v1=$_POST['txt_v1'];
$v2=$_POST['txt_v2'];
$BTN=$_POST['btn_calc'];
If ($BTN!=""){
    If ($v1==' ' or $v2==' '){ //VERIFICANDO SE AS CAIXAS ESTÃO VAZIAS
        echo "ATENÇÃO! UM DOS VALORES NÃO FOI INFORMADO!";
    }
    Else{
        $total=$v1+$v2;
        echo "A soma dos dois valores é:$total";
    }
}
?>
</center>
</body>
```

ATENÇÃO! Pode acontecer de parecer estar tudo certo e o PHP se comporta como se algum campo estivesse vazio, mesmo com as caixas preenchidas. Verifique com cuidado as variáveis (minúsculo ou maiúsculo) em todo o código, pois podem estar diferentes e deixe-as iguais ou se você se esqueceu de usar == na expressão.

3. FORMATAR VALORES EM ESTILO MOEDA (R\$)

Podemos usar o `number_format` para que o PHP imprima valores monetários já formatados, seguindo o esquema a seguir:

```
number_format($variavel_com_numero, int casa decimais, separador decimal , separador de milhares);
```

Exemplo:

```
<?php
    $num = 100000.50;
    // REPARA QUE O PADRÃO É NO FORMATO AMERICANO
    echo 'R$' . number_format($num, 2); //RETORNA R$100,000.50
    //NOSSO FORMATO
    echo 'R$' . number_format($num, 2, ',', '.'); //RETORNA R$100.000,50
    //FORMATO AMERICANO
    echo 'R$' . number_format($num, 2, '.', ','); //RETORNA R$100,000.50
?>
```

Alteraremos então o código do nosso arquivo **form_calcula.php** aplicando o estilo de moeda em nosso formato nos resultados apresentados:

```
<html>
<head><title>Exemplo Form-Híbrido</title></head>
<body><center>
<form action='form_calcula.php' method='post'>
    Valor 1
    <input type='text' name='txt_v1'><br>
    Valor 2
    <input type='text' name='txt_v2'><br>
    <input type='submit' name='btn_calc' value='Calcular'><br>
    <?php
        $v1=$_POST['txt_v1'];
        $v2=$_POST['txt_v2'];
        $BTN=$_POST['btn_calc'];
        If ($BTN!=""){
            If ($v1=="" or $v2==""){
                echo "ATENÇÃO! UM DOS VALORES NÃO FOI INFORMADO!";
            }
            Else{
                $total=$v1+$v2;
                echo "A soma dos dois valores é: R$".number_format($total, 2, ',', '.');
            }
        }
    ?>
</center>
</body>
```

4. CAIXAS DE MENSAGENS COM JAVASCRIPT

Para criar caixas de mensagens, utilizaremos um comando JavaScript dentro do PHP. Este comando pode ser utilizado mensagem inicial, como uma saudação, para emitir alertas aos usuários do site ou em um tratamento de erro, substituindo o echo. Veja o exemplo:

```
<?php
```

//USAR ECHO, POIS O PHP TAMBÉM NÃO CONHECE JAVASCRIPT (NÃO FECHAR ASPAS AINDA!)

```
echo "
<script language=javascript>
alert('ISTO É UMA CAIXA DE MENSAGEM!! GOSTOU??');
</script>
";
//AQUI FECHAMOS AS ASPAS DO ECHO
?>
```

Com este tratamento, nosso **form_calcula.php** ficaria assim:

```
<html>
<head><title>Exemplo Form-Híbrido</title></head>
<body><center>
<form action='form_calcula.php' method='post'>
Valor 1
<input type='text' name='txt_v1'><br>
Valor 2
<input type='text' name='txt_v2'><br>
<input type='submit' name='btn_calc' value='Calcular'><br>
<?php
$V1=$_POST['txt_v1'];
$V2=$_POST['txt_v2'];
$BTN=$_POST['btn_calc'];
If ($BTN!=""){
    If ($V1=="" or $V2==""){
        echo "ATENÇÃO! UM DOS VALORES NÃO FOI INFORMADO!";
    }
    Else{
        $total=$V1+$V2;
        echo " <script language=javascript>
        alert('CÁLCULO EFETUADO!!');
        </script>";
        echo "A soma dos dois valores é: R$".number_format($total, 2, ',', '.');
    }
}
?>
</center>
</body>
```

5. VALIDAR E-MAIL

Para verificarmos se certo dado é um e-mail ou não é comum utilizarmos **expressões regulares** e funções como `preg_match()`. Neste tutorial de hoje iremos aprender uma função nativa do `php` para **validar e-mail** sem sacrifício.

A vantagem de utilizar essa solução é que seu script fica bem menor e, principalmente, não fica confuso, com expressões regulares que muitas vezes não entendemos. Como o `php` tem uma função para validação de e-mail, vamos utilizá-la.

A validação de e-mail com expressão regular ficaria parecida com esta:

```
<?php  
$email = "emailteste@meusite.com";  
if (preg_match ("^/[A-Za-z0-9]+([_.-][A-Za-z0-9]+)*@[A-Za-z0-9]+([_.-][A-Za-z0-9]+)*\\.[A-Za-z0-9]{2,4}$", $email)) {  
    echo "O e-mail é válido!";  
} else {  
    echo "O e-mail é inválido!";  
}  
?>
```

Podemos perceber que esta função não é a mais adequada, necessita de algum conhecimento de expressão regular e ainda pode causar erros. Tente validar o seguinte e-mail `teste_@meusite.com` com esta função . O resultado vai ser e-mail inválido, isto por conta do `_` antes do `@`, mas este e-mail pode muito bem existir.

Então como validar e-mail facilmente com `php`?

Vamos utilizar a função `filter_var()`, veja:

```
<?php  
$email = 'teste@meusite.com';  
if(filter_var($email, FILTER_VALIDATE_EMAIL)){  
    echo 'E-mail válido';  
} else{  
    echo 'E-mail invalido';  
}  
?>
```

IMPORTANTE! Esta função só funciona com `php 5` ou superior.

Dica tirada de:

http://www.comocriarsites.com.br/Como-validar-e-mail-com-php-sem-usar-expressao-regular_b_16.html

Agora aproveitamos esta dica e utilizamos para tratar um formulário, como no exemplo a seguir:

```
<html>

<head><title>Valida e-mail fácil</title></head>

<body>

<center>

<form action="valida_mail_facil.php" method="post" name="emailForm">

Email: <input type="text" name="txt_email"><br/>

<input type="submit" name="submitemail" value="Verificar">

</form>

<?php

    $email = $_POST['txt_email'];

    if(filter_var($email, FILTER_VALIDATE_EMAIL)){

        echo "O e-mail $email é válido";

    }else{

        echo ' O e-mail $email é inválido';

    }

?>

</center>

</body>

</html>
```

ATIVIDADES (TRATAMENTO DE FORMULÁRIOS):

Refaça os dois exercícios anteriores (com código híbrido e separado) e aplique todos os tratamentos de erros necessários.

FORMULÁRIO COM O NVU

O NVU é um programa gratuito, para criação e edição de sites em modo visual e HTML.

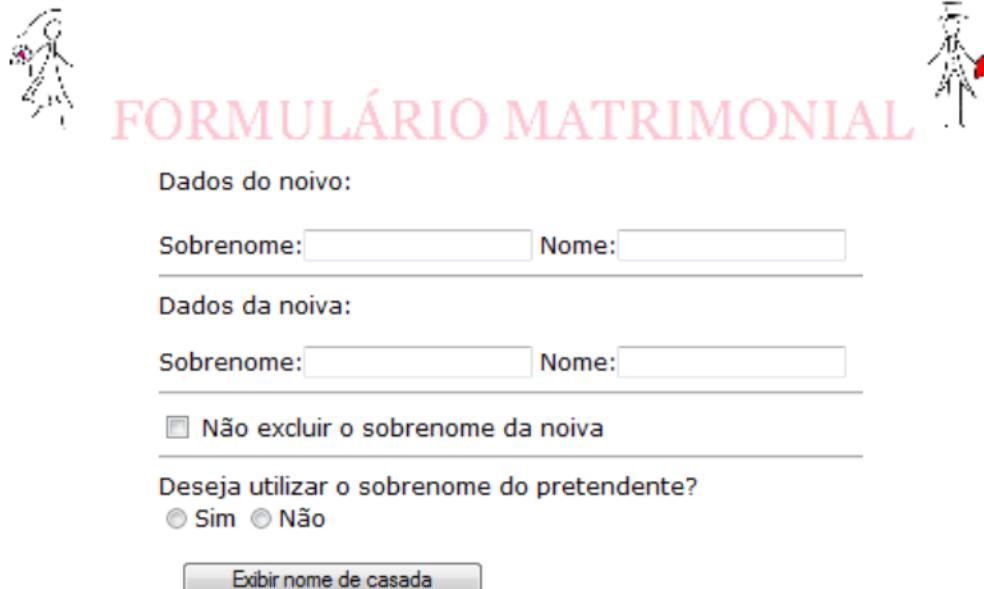
Trata-se de um programa tão completo quanto se trata de desenvolvimento de sites que ele chega a ser comparável até mesmo com o FrontPage da Microsoft ou o Adobe Dreamweaver, que são programas pagos.

Assim como qualquer outro programa deste tipo ele permite inserir e editar textos, inserir imagens, links, formulários, tabelas e células.

ATENÇÃO! O NVU não foi desenvolvido para trabalhar com linguagens dinâmicas como o PHP e ASP. Sempre que tentar abrir um arquivo deste tipo ele oferecerá a opção de abrir o arquivo com outro programa.

Utilizaremos o NVU apenas para criar o visual dos formulários híbridos ou separados e para a programação em PHP continuaremos usando o Notepad++.

Como exemplo, faremos um formulário que montará o nome de casado da noiva de acordo com os dados e escolhas fornecidos. Observe a figura a seguir:



The screenshot shows a web-based form titled "FORMULÁRIO MATRIMONIAL". The form is designed to merge the bride's and groom's names. It includes fields for both surnames and first names, a checkbox for "Não excluir o sobrenome da noiva" (Do not remove the bride's surname), and a radio button section for "Deseja utilizar o sobrenome do pretendente?" (Do you want to use the groom's surname?). A button at the bottom says "Exibir nome de casada" (Show married name). There are also small icons of a bride and groom on either side of the title.

Dados do noivo:

Sobrenome: Nome:

Dados da noiva:

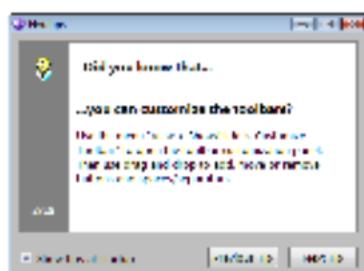
Sobrenome: Nome:

Não excluir o sobrenome da noiva

Deseja utilizar o sobrenome do pretendente?

Sim Não

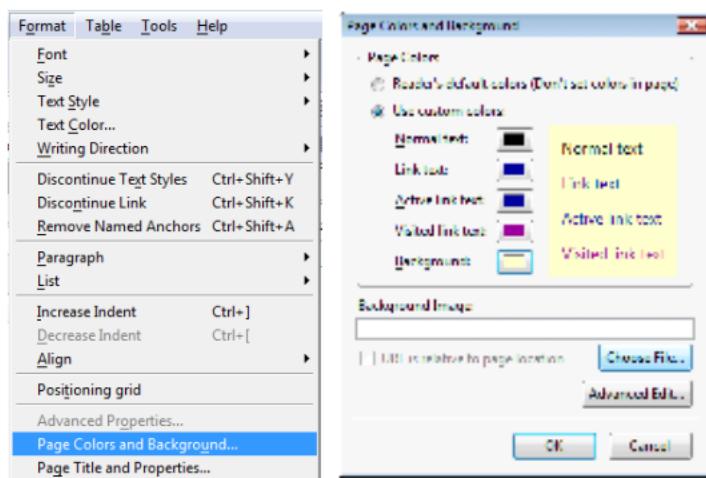
Começaremos abrindo o NVU.



A janela NVU Tips (dicas NVU) pode ser fechada.

Caso você não deseje que ela se abra sempre que iniciar o NVU, desmarque a opção Show tips at startut.

Se quiser modificar as cores padrão para o fundo da página ou para o texto clique em Format, Page Colors and Background:

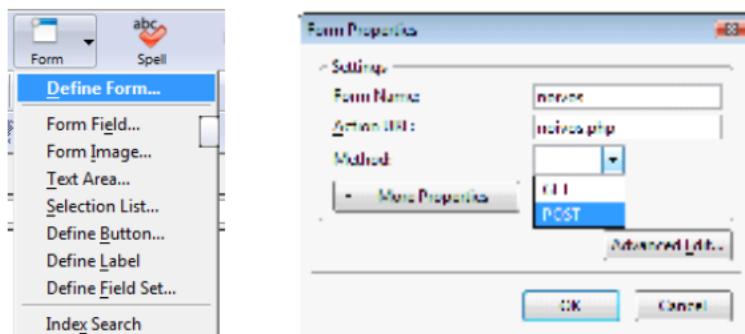


Clique na opção Use custom colors e selecione as cores desejadas.

Se quiser uma imagem no fundo (Não recomendamos), na opção Background Image, clique em Choose File e escolha sua imagem.

Ao concluir estes ajustes, clique em OK.

Assim como no código escrito, a primeira coisa fazer é declarar um formulário e passar os parâmetros Method e Action. Clique no botão Form e escolha a opção Define Form...



Digite um nome para o formulário (Opcional).

O Action URL é sempre o nome do PHP.

Escolha o método (Neste exemplo o POST)

Um espaço pontilhado indica o formulário criado.



ATENÇÃO! Tudo o que fizer parte do formulário deverá estar dentro deste espaço pontilhado. Se você fizer as caixas de texto, botões e outros objetos fora deste espaço o PHP não funcionará.

Apenas por uma questão de estética criaremos primeiro uma tabela, para organizar melhor os objetos de nosso formulário.

O formulário que montaremos tem 7 linhas e 2 colunas. Clique no botão Table

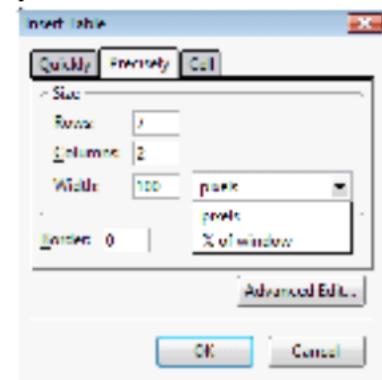
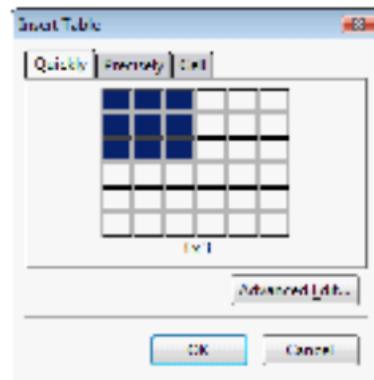


Podemos dimensionar a tabela como fazemos no Word, da Microsoft. Se quisermos digitar a quantidade de linhas e colunas e ainda modificarmos a borda da tabela, clicamos em Precisely

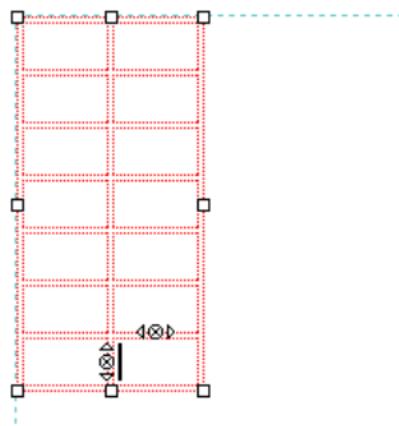
Digitamos a quantidade de linhas (**Rows**) e colunas (**Columns**).

Deixaremos a borda (**Border**) com 0, pois queremos a tabela “invisível”.

Em **Width** definimos o tamanho que a tabela ocupará na tela. **Pixels** é tamanho fixo e **% of window** é tamanho baseado na resolução do monitor.



Clicando em OK, veremos nossa tabela na tela do NVU. Como nossa tabela tem borda 0, somente veremos a tabela no NVU. A página não mostrará a tabela quando estivermos navegando.



Abaixo do NVU temos 4 guias:

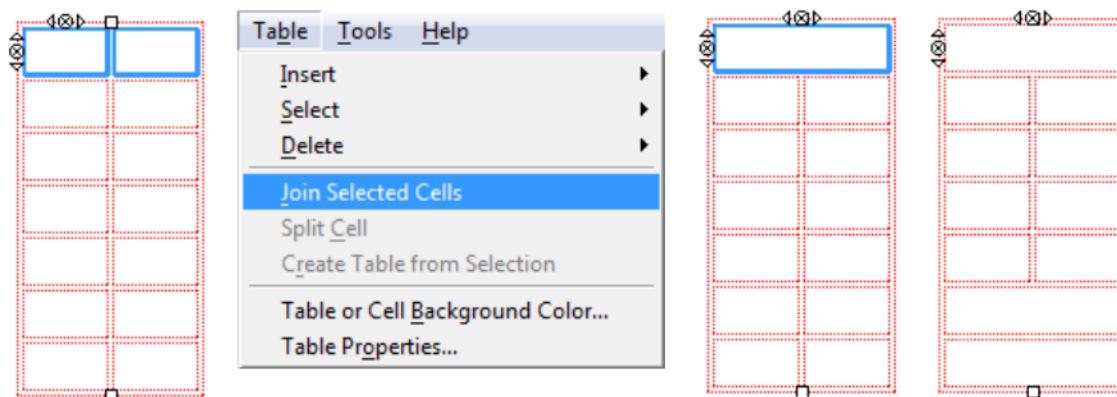
- **Normal:** A tela de trabalho do NVU. Aqui vemos tudo o que está sendo feito, inclusive tabelas invisíveis (com borda 0).
- **HTML Tags:** Mostra caixas com os comandos HTML de cada objeto
- **Source:** Mostra o código HTML que é gerado.
- **Preview:** Para ver como ficaria a página pronta.

Clique em Preview e veja que a tabela não aparece. Como comentado antes, a tabela está aí apenas para organizar os objetos do formulário. Clique em Normal para continuar.

Para centralizar a tabela, clique dentro dela (em qualquer parte) e observe que na parte de baixo do NVU aparecem os comando <body><form><table><tbody><tr><td>. Clique em <table> e no botão centralizar.



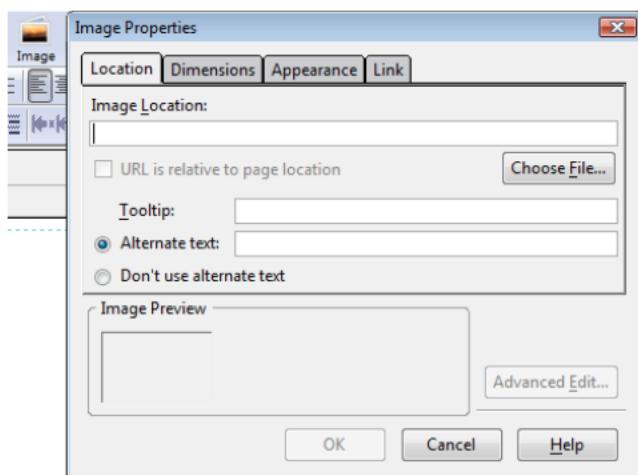
Agora mesclaremos a primeira linha da tabela. Clique dentro da primeira célula e arraste para a segunda. Ambas ficarão com um contorno azul. Depois iremos no menu **Table**, **Join Selected Cells**. Repetiremos isto nas duas últimas linhas da tabela.



ATENÇÃO! Para ajustar a largura das colunas da tabela você deve utilizar a régua que fica na parte superior da tela do NVU.



Na primeira linha colocaremos uma imagem. Clique no botão **Centralizar** e depois em **Image** e em **Choose file**



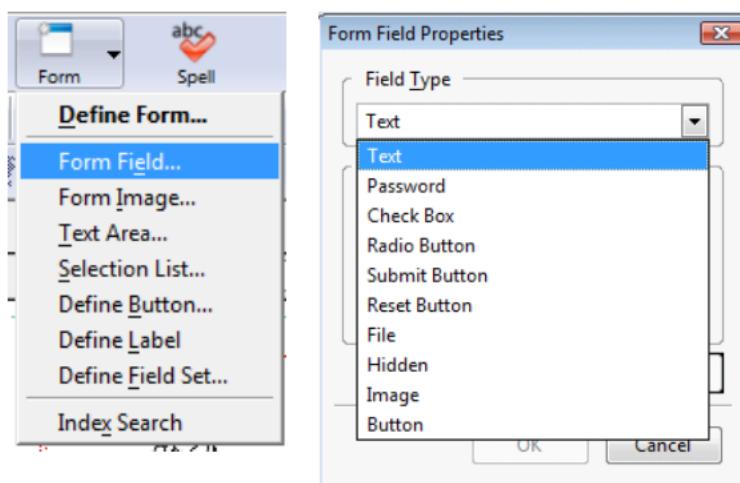
Basta procurar e selecionar a imagem (gif, jpg ou png) e clicar no botão OK.

Vale lembrar que as imagens usadas aqui têm que estar gravadas no computador e na mesma pasta do projeto.

Alternative text é usado para caso sua imagem não consiga ser carregada.

Colocaremos uma imagem na esquerda, um título e uma imagem à direita do título, ficando assim e colocaremos os textos para sabermos que campos serão preenchidos:

E finalmente criaremos os campos e botões de nosso formulário, começando pela caixa de texto que receberá o Sobrenome do Noivo. Posicione o cursor nesta coluna, depois do texto e clique no botão Form, opção Form Field:



Temos aqui todos os objetos que compõe um formulário:

Text: Caixas de texto comuns

Password: Para senhas

Check Box: Caixa de escolha
(Várias escolhas)

Radio Button: Botões “bolinha”
para uma escolha

Submit Button: Botão que ativa
o PHP

Reset Button: Botão que limpa
os campos

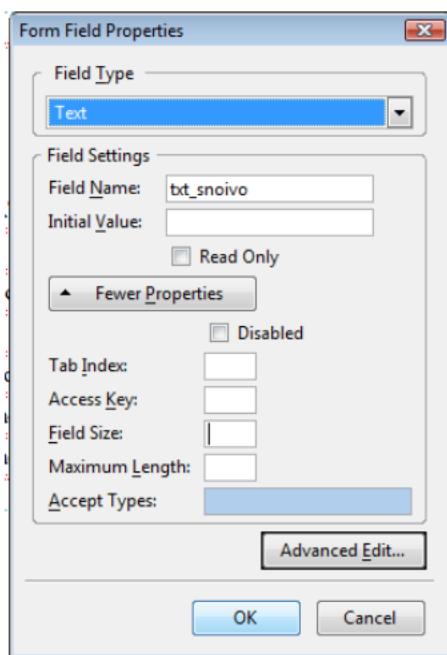
File: Botão para anexar
arquivos

Hidden: Comandos ocultos

Image: Imagens tipo botão

Button: Outro tipo de botão

Como queremos que o usuário digite a informação, precisamos de uma Text:



Em Field Name daremos o nome de nossa caixa de texto, que será **txt_snoivo** (sobrenome do noivo).

O botão **More Properties** abre opções avançadas para a caixa de texto e as mais usadas são:

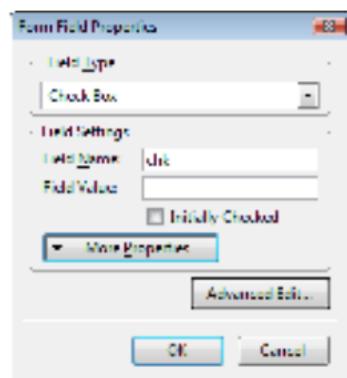
- **Filed Size:** Tamanho do desenho da caixa de texto
- **Maximum Length:** Quantos caracteres a caixa aceita

Podemos ter, por exemplo, uma caixa para CEP de tamanho 10, mas que aceita no máximo 9 caracteres.

Criaremos ainda **txt_nnoivo** (nome do noivo), **txt_snoiva** (sobrenome da noiva) e **txt_nnoiva** (sobrenome da noiva).

Nosso form terá todos os campos para digitação, faltando apenas os checkbox e radio buttons para as demais opções:

Posicionaremos o cursor na linha 6, antes da opção “**Não excluir o sobrenome da noiva**” e adicionaremos um novo Form Field, escolhendo agora a opção **Check Box**:

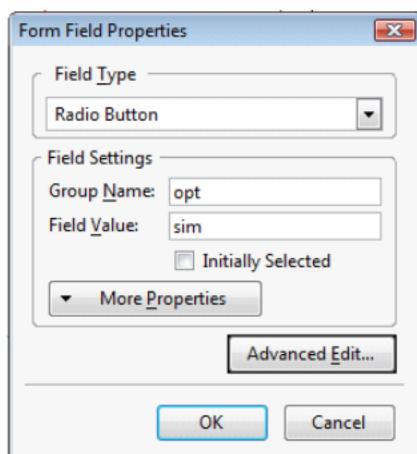


A Check Box se chamará chk e servirá para que a pessoa escolha entre manter ou não o sobrenome de solteiro da noiva.

Se ativado, não excluirá o sobrenome, juntando com o sobrenome do noivo dependendo da opção abaixo.

Se desativado, o sobrenome de solteiro será excluído automaticamente e o do noivo poderá se o único sobrenome da noiva.

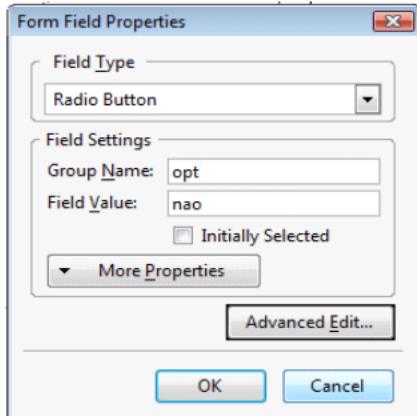
Na última linha, após a pergunta “Deseja utilizar o sobrenome do pretendente?” adicionaremos um Radio Button em Form Field, escolhendo agora a opção Radio Button:



Coloque em **Group Name** o nome opt e em **Field Value** o valor sim

Clique em OK, mas cuidado! Só aparecerá o radio, ou seja, a bolinha. Você deve ainda escrever na frente do botão o texto que o usuário usará como base para a escolha, no caso a palavra Sim.

Se isso não for feito, o usuário não saberá o que está escolhendo.



Coloque em **Group Name** o nome opt e em **Field Value** o valor não (sem acento)

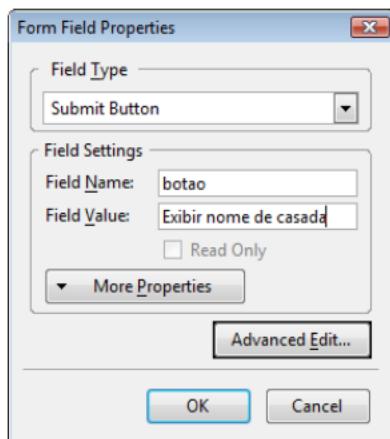
Clique em OK, mas cuidado! Só aparecerá o radio, ou seja, a bolinha. Você deve ainda escrever na frente do botão o texto que o usuário usará como base para a escolha, no caso a palavra Não.

Se isso não for feito, o usuário não saberá o que está escolhendo.

ATENÇÃO! Perceba que seja a opção é Sim ou Não seu **Group Name** será o mesmo! Como fazem parte uma opção de escolha para um único assunto, devem ter o mesmo nome, para garantir que a pessoa escolha Sim ou Não. Caso este cuidado não seja tomado, você permitirá que o usuário escolha Sim e Não na mesma opção. Esteja a vontade para testar este erro se quiser.

Abaixo colocaremos o botão que se encarregará de ativar o PHP quando for clicado, enviando as informações seguindo os critérios de escolha de método.

Clique novamente no botão Form e em Form Field, escolhendo agora a opção **Submit Button**:



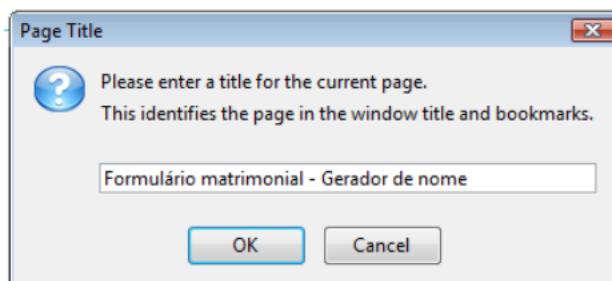
Cuidado para não confundir Submit Button com Button (última opção), pois são objetos diferentes.

Preencha Field Name com botao e Field Value com Exibir nome de casada

Clique em OK e veja o resultado

Finalmente nosso formulário está pronto e é chegada a hora da gravação. Clique em File, opção Save:

Surgirá uma janela pedindo o Page Title (Título da Página). Este **não é o nome da página** e sim o que aparecerá na Barra de Título do navegador e na Barra de Tarefas do Windows.



Clique em OK para confirmar o título e ir para a tela de gravação. Nossa trabalho com o NVU termina aqui.

CUIDADO! O nome do arquivo depende de que tipo de código você está fazendo. Se for um código híbrido tem que ser o mesmo nome colocado em Action quando criamos o formulário. Se for código separado fica a seu critério, desde que respeitando as regras de nome. No nosso caso, será um código híbrido, portanto **noivos.php**.

Abriremos o código do formulário gerado, clicando no símbolo do wamp, opção www directory. Procure a pasta do projeto atual e clique com o botão direito do mouse, escolhendo a opção Edit with Notepad++. Adicionaremos o código PHP no final do código gerado:

```
<html>
<head>
<title> Formulário Matrimonial </title>
</head>
<body >
<form action='noivos.php' method='post'>
<center><img src='noivinha.jpg'widht=50 height=120 align=left>
<font color='pink' size='8' face='georgia'>FORMULÁRIO MATRIMONIAL </font>
<img src='noivo.jpg'widht=50 height=120></center>
<br><br><br>
<font face='verdana' size='2'>
Dados do noivo:
Sobrenome:<input type='text' name='txt_snoivo'>
Nome:<input type='text' name='txtn_noivo'>
<br>
<hr>
Dados da noiva:<br><br>
Sobrenome:<input type='text' name='txt_snoiva'>
```

Nome:<input type='text' name='txt_nnoiva'>

<hr>

<input type='checkbox' value=s name='chk'> Não excluir o sobrenome da noiva

<hr>

Deseja utilizar o sobrenome do pretendente?

<input type='radio' value='sim' name='opt'>Sim

<input type='radio' value='não' name='opt'>Não

<input type='submit' value='Exibir nome de casada' name='botao'>

</form>

</body>

</html>

<?php

```
$nomenoiva=$_POST['txt_nnoiva'];

$sobrenoiva=$_POST['txt_snoiva'];

$nomenoivo=$_POST['txt_nnoivo'];

$sobrenomenoivo=$_POST['txt_snoivo'];

$botao=$_POST['botao'];

$opt=$_POST['opt'];

$chk=$_POST['chk'];

$resu='O nome de casado da pretendente será: ';

if($botao!=""){


```

```
if(($nomenoiva=="")||($sobrenoiva=="")||($nomenoivo=="")||($sobrenoivo=="")){  
    echo'<font color=red>';  
    echo 'Atenção!Preencha todos os campos!';}  
  
else{  
  
    if(($opt=='sim') && ($chk!="")){  
  
        echo  
$resu.$nomenoiva.'&nbsp;'.$sobrenoiva.'&nbsp;'.$sobrenomenoivo;}  
  
    elseif (($opt=='sim') && ($chk=="")) {  
  
        echo $resu.$nomenoiva.'&nbsp;'.$sobrenomenoivo;}  
  
    else{  
  
        echo $resu.'&nbsp;'.$nomenoiva.'&nbsp;'.$sobrenoiva;}  
  
    }  
  
}  
  
?>
```

ATIVIDADES (NVU):

1. Refaça os dois exercícios anteriores usando o NVU (com código híbrido e separado) e aplique todos os tratamentos de erros necessários.

OBS: O primeiro exercício (cálculo de salário) deverá ainda ter uma pergunta “FUNCIONÁRIO TEM FALTA?” com dois Radio Buttons SIM e NÃO.

Caso a resposta seja sim, deverá ser descontado do salário líquido mais 4% como punição.

2. Faça um formulário que receba o nome do funcionário, seu salário bruto e a quantidade de meses trabalhados. Neste formulário deverá ter ainda dois Radio Buttons PARCELAR e NÃO PARCELAR. O PHP deverá calcular o 13º salário a ser pago (DIVIDIR O SALÁRIO POR 12 E MULTIPLICAR PELA QUANTIDADE DE MESES TRABALHADOS) e verificar se haverá parcelamento ou não. Caso haja, o 13º será dividido em duas parcelas iguais. (com código híbrido e separado)

PHP COM BANCO DE DADOS

O PHP MYADMIN

PHPMyAdmin é um programa de computador desenvolvido em PHP para administração do MySQL pela Internet. A partir deste sistema é possível criar e remover bases de dados, criar, remover e alterar tabelas, inserir, remover e editar campos, executar códigos SQL e manipular campos chaves.

USANDO O PHP MYADMIN

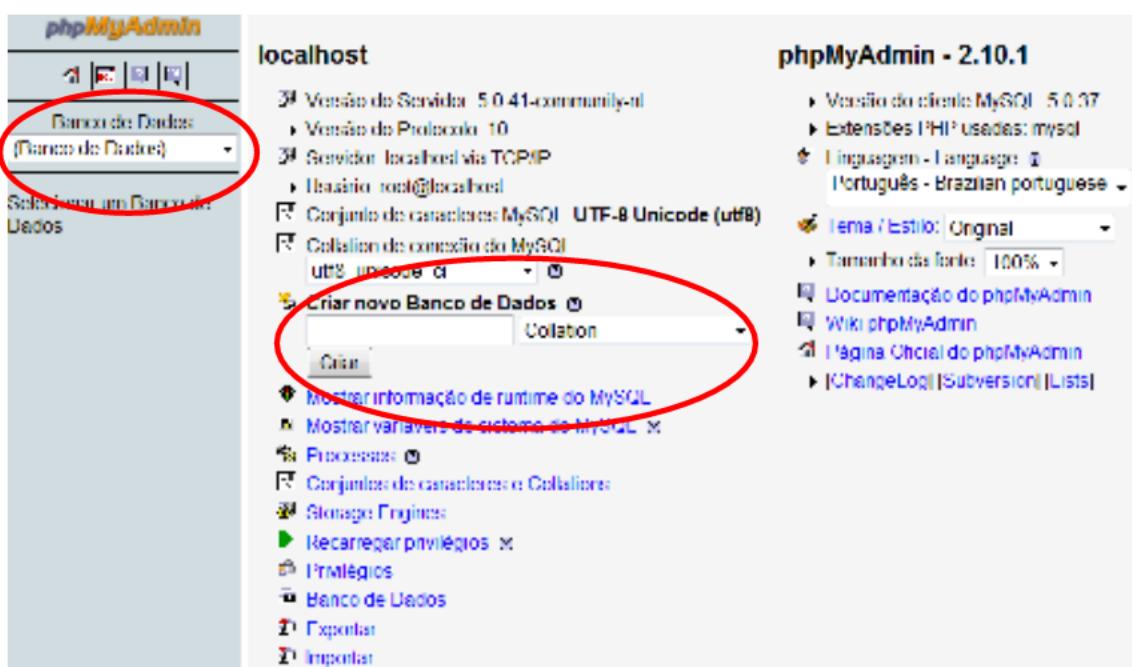
A abertura do PHP MyAdmin é feita da mesma forma que a abertura de uma página em PHP, ou seja, pelo ícone do wamp  , opção **php MyAdmin**.

www.proralph.com.br

ATENÇÃO! Podemos ter aqui o problema citado anteriormente, onde seu navegador pode abrir um site de endereço **www.localhost.com.br**. Se este problema acontecer, vá à barra de endereços do seu navegador e substitua este endereço por **//127.0.0.1**, assim, será carregada a página do Wamp pelo ip da sua placa de rede. Clique no link do **php MyAdmin** e você poderá trabalhar normalmente.

A tela do php MyAdmin é bem clara, organizada e intuitiva. Com um conhecimento mínimo de Banco de Dados é possível trabalhar com ela sem problemas. No canto esquerdo temos a opção Banco de Dados, onde podemos escolher qual banco criado queremos abrir e no lado direito (o maior) há um local para digitarmos o nome do banco que queremos criar, além de opções de configurações.

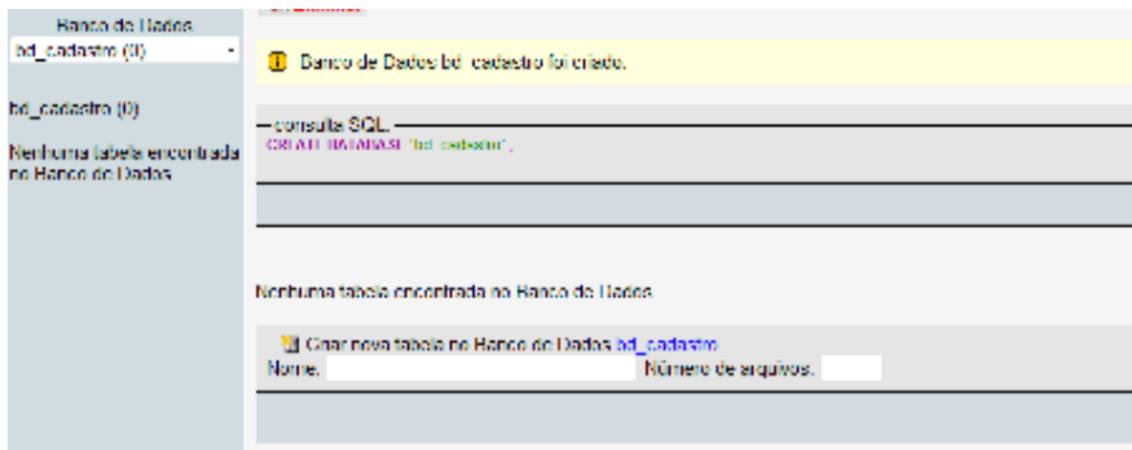
Basicamente você só precisa saber criar e acessar seu banco de dados.



The screenshot shows the phpMyAdmin interface for the 'localhost' server. In the top right corner, there's a dropdown menu set to 'Português - Brazilian portuguese'. Below it, the 'Tema / Estilo' is set to 'Original'. The main left sidebar has a red circle around the 'Banco de Dados' section, which contains a dropdown menu with 'Criar novo Banco de Dados'. This dropdown also has a red circle around it. The 'Criar' button is highlighted with a red circle.

Para conhecer o PHP MyAdmin vamos criar um banco de dados que será utilizado posteriormente em conjunto com um formulário de cadastro.

Vamos criar um banco de dados chamado `bd_cadastro`. Basta digitar o nome no local indicado e clicar no botão **Criar**. Se tudo correr bem, teremos uma mensagem de confirmação:



The screenshot shows the 'Banco de Dados' section of phpMyAdmin. On the left, under 'Banco de Dados', there is a list with 'bd_cadastro (0)' selected. To its right, a message says 'Banco de Dados bd_cadastro foi criado.' Below this, a SQL query is shown: 'CREATE DATABASE `bd_cadastro`';. At the bottom, there is a form to 'Criar nova tabela no Banco de Dados bd_cadastro' with 'Nome:' set to 'Nome' and 'Número de arquivos:' set to '1'.

Podemos reparar que agora, no canto esquerdo da tela há na lista de escolha e que nosso Banco de Dados criado consta nesta lista e ao lado um número 0 entre parênteses. Significa que como não temos ainda nenhuma tabela é mostrado `bd_cadastro(0)`, ou seja, entre parênteses fica o número de tabelas que nosso banco possui.

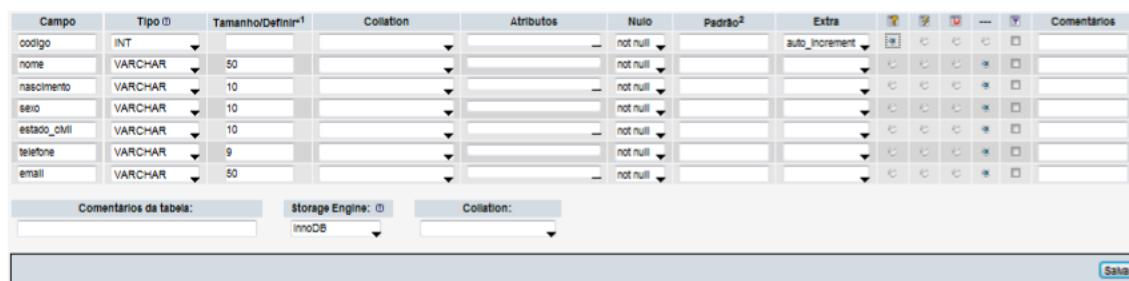
Temos ainda no meio da página um espaço (consulta SQL) que mostra qual é o comando SQL para se criar um banco de dados (ou seja, PHPMyAdmin também é ótimo para aprender comandos SQL) e abaixo deste espaço, há um local para a criação de nossas tabelas, onde Nome: é o nome da

nossa tabela e Número de arquivos: é quantos campos nossa tabela terá (temos aqui uma tradução mal feita).

Criaremos uma tabela chamada tb_dados para cadastrar codigo, nome, nascimento, sexo, estado_civil, telefone e email, ou seja, 7 arquivos (campos).

Basicamente preencheremos nome dos campos, tipos de dados, tamanho dos campos, definiremos a chave primária e faremos com que a numeração seja automática para o código.

Nome, Tipo e Tamanho são as três primeiras colunas a preencher. Mudaremos a opção Extra do campo codigo para Auto_increment e a próxima opção já é a chave primária.

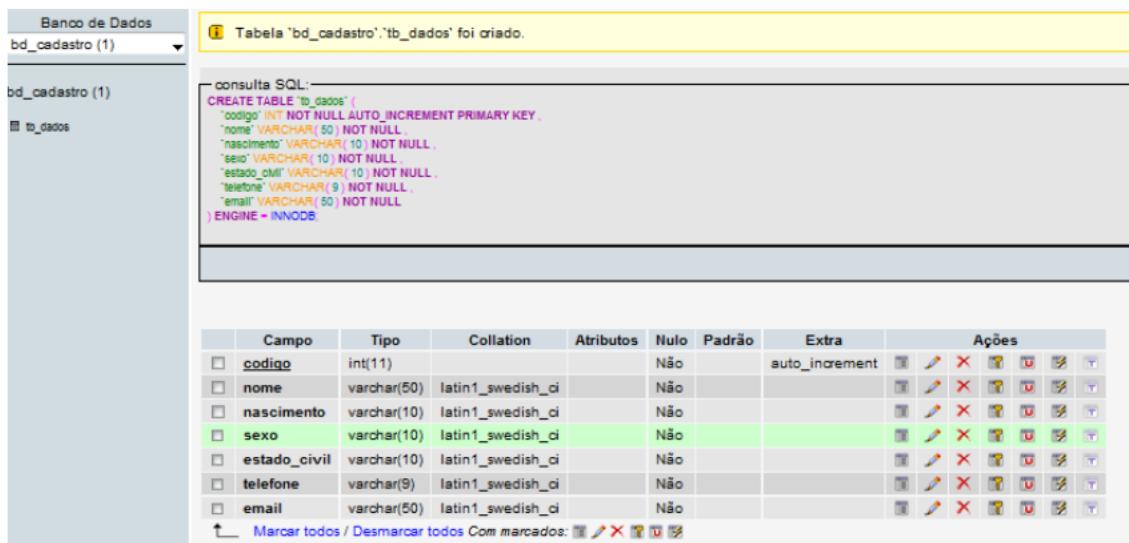


A screenshot of the MySQL Workbench interface showing the creation of a new table. The table structure is as follows:

| Campo | Tipo | Tamanho/Definir ¹ | Collation | Atributos | Nulo | Padrão ² | Extra | Comentários |
|--------------|---------|------------------------------|-----------|-----------|----------|---------------------|----------------|-------------|
| codigo | INT | | | | not null | | auto_increment | |
| nome | VARCHAR | 50 | | | not null | | | |
| nascimento | VARCHAR | 10 | | | not null | | | |
| sexo | VARCHAR | 10 | | | not null | | | |
| estado_civil | VARCHAR | 10 | | | not null | | | |
| telefone | VARCHAR | 9 | | | not null | | | |
| email | VARCHAR | 50 | | | not null | | | |

Below the table structure, there are buttons for 'Comentários da tabela:', 'Storage Engine:', and 'Collation:', with 'InnoDB' selected for Storage Engine and 'latin1_swedish_ci' for Collation. A 'Salvar' (Save) button is located at the bottom right.

Ao terminar clicaremos em Salvar



A screenshot of the MySQL Workbench interface showing the results of the table creation. The message area says 'Tabela 'bd_cadastro'.'tb_dados' foi criado.' (Table 'bd_cadastro'.'tb_dados' was created). Below it, the SQL query used to create the table is shown:

```
CREATE TABLE `tb_dados` (
  `codigo` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `nome` VARCHAR(50) NOT NULL,
  `nascimento` VARCHAR(10) NOT NULL,
  `sexo` VARCHAR(10) NOT NULL,
  `estado_civil` VARCHAR(10) NOT NULL,
  `telefone` VARCHAR(9) NOT NULL,
  `email` VARCHAR(50) NOT NULL
) ENGINE = INNODB;
```

The left sidebar shows the database 'bd_cadastro (1)' and the table 'tb_dados'. The main pane displays the table structure with columns: Campo, Tipo, Collation, Atributos, Nulo, Padrão, Extra, and Ações (Actions). The 'Extra' column for the 'codigo' field shows 'auto_increment'. The 'Ações' column for the 'sexo' field has a green highlight.

Novamente podemos reparar o código SQL, abaixo a estrutura da tabela e no canto esquerdo agora vemos **bd_cadastro(1)** que indica que temos uma tabela, mostrada logo abaixo (**tb_dados**). Caso a tabela estivesse com dados cadastrados, poderíamos vê-los clicando no nome da tabela.

Nosso banco de dados está criado e pronto para ser usado pelo PHP.

PHP INTERAGINDO COM O SGDB MYSQL

Nesta seção serão apresentadas as principais funções, ou rotinas, PHP utilizadas para interagir com o SGDB MySQL.

COMO PHP INTERAGE COM O SGDB MYSQL

PHP fornece uma série de funcionalidades para a interação com o SGDB MySQL. A seqüência lógica dessa interação pode ser descrita pelos seguintes passos:

- **Primeiro passo:** estabelecer uma conexão com o SGDB MySQL.
- **Segundo passo:** selecionar o banco de dados utilizado.
- **Terceiro passo:** executar a SQL desejada.
- **Quarto passo:** fechar a conexão.

COMO ABRIR UMA CONEXÃO

Uma forma de se estabelecer a conexão com SGDB MySQL é utilizando a função **mysql_pconnect()**. Esta função abre uma conexão persistente com o SGDB. Ela retorna um link (uma ligação) caso a conexão tenha sido estabelecida ou retorna false caso a conexão não pôde ser estabelecida. Existem várias assinaturas para esta função, no entanto uma bem comum é aquela onde são passados o **host**, o **usuário** e o **password**. A assinatura descrita segue abaixo:

```
$conexao = mysql_pconnect('host', 'usuario', 'senha')
```

Onde **\$conexao** é a variável que conterá o link caso a conexão tenha sido estabelecida, host é o local onde o SGBD está instalado e provendo o serviço. Usuário e senha são, respectivamente, o usuário e sua senha no SGDB.

COMO SELECIONAR O BANCO DE DADOS

Uma vez criado o link com o SGDB deve-se indicar qual o banco de dados que será trabalhado. Para tal faz-se uso da função **mysql_select_db()**. Ela retorna true ou false dependendo se ela conseguiu ou não ativar o banco de dados indicado. Uma assinatura possível é aquela onde são passados o banco e o link. A assinatura descrita segue abaixo:

```
$conseguiu = mysql_select_db('nome_banco', $conexao)
```

COMO EXECUTAR UMA SQL

Para executar uma SQL no SGDB MySQL utiliza-se a função **mysql_query()**. Esta função retorna um ResultSet contendo os resultados da consulta caso a query seja um **SELECT** ou então ela retorna true ou false se a query for **INSERT**, **DELETE** ou **UPDATE**. Uma assinatura possível é aquela onde são passados a query e o link. A assinatura descrita segue a seguir:

```
$resultado = mysql_query($sql, $conexao)
```

Onde **\$sql** é uma variável que contém uma string com a operação desejada.

COMO MANIPULAR O RESULTSET DA EXECUÇÃO ANTERIOR

A função **mysql_query()** retorna, como mencionado, um ResultSet. Talvez uma das maneiras mais simples de se manipular esse ResultSet é transformando-o em uma matriz MxN onde M é o número de linhas contidas no resultado da consulta e N o número de colunas contidas no resultado da consulta.

Para obter tal resultado faremos uso da função **mysql_fetch_row()**. Esta função retorna um array 1xN contendo os valores de uma tupla e posiciona o ponteiro do resultSet na próxima tupla. O parâmetro dessa função é uma variável contendo o resultSet.

Segue abaixo um exemplo dessa transformação de um resultSet em uma matriz:

```
$resultSet = mysql_query("SELECT nome, rg FROM tabela_pessoa", $conexao)
```

```
$l = 0;
```

```
while($linha = mysql_fetch_row($resultSet)){ // para passar por todas as linhas
```

```
    $resultado[$l] = $linha;
```

```
    $l++;
```

```
}
```

Depois dessa execução, a variável resultado será uma matriz de Mx2, onde M é o número de tuplas da tabela_pessoa. Cada posição **\$l,\$c** corresponde à linha de índice **\$l** (começando de zero) e à coluna de índice **\$c** do resultSet (começando de zero). Por exemplo, **\$resultado[0][0]** conterá a string “Fulano da Silva” e **\$resultado[0][1]** a string “R01”.

COMO FECHAR A CONEXÃO

Para fechar a conexão aberta no item 2.b é utilizada a função **mysql_close()**. Ela retorna true ou false dependendo se ela conseguiu ou não fechar a conexão. Como parâmetro, geralmente é passado o link aberto quando fizemos a conexão. A sintaxe é a seguinte:

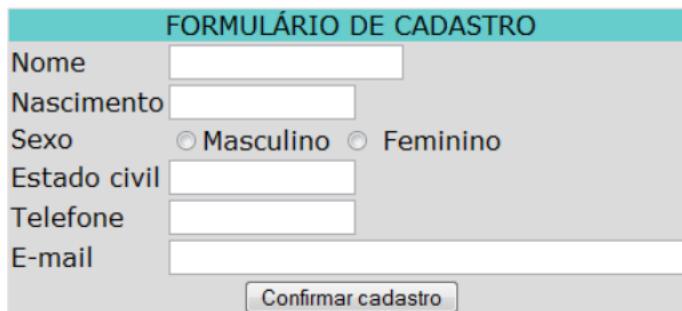
```
$conseguiu = mysql_close($conexao)
```

MANIPULAÇÃO DO BANCO DE DADOS

Manipular (gravar, alterar, pesquisar, apagar) informações no nosso banco de dados utilizando PHP com comandos SQL é um processo relativamente simples como veremos nesta parte de nossa apostila.

CADASTRO (GRAVAÇÃO)

Criaremos um formulário com os mesmos campos criados em nosso banco de dados (com exceção do código, que é auto_increment). Para este trabalho, utilizaremos novamente o NVU e criaremos o formulário a seguir, chamado **cadastro.php**.



FORMULÁRIO DE CADASTRO

Nome

Nascimento

Sexo Masculino Feminino

Estado civil

Telefone

E-mail

Confirmar cadastro

Abriremos o código gerado e faremos o código PHP que receberá os dados e gravará no banco de dados.

Acrescente no final do código o script PHP abaixo:

```
<?php  
  
//RECEBEMOS OS DADOS DO FORM  
$nome=$_POST['txtnome'];  
  
$nascimento=$_POST['txtnasc'];  
  
$sexo=$_POST['optsexo'];  
  
$civil=$_POST['txtcivil'];  
  
$fone=$_POST['txtfone'];  
  
$email=$_POST['txtmail'];  
  
$enviar=$_POST['btncadastra'];  
  
//DADOS DA CONEXÃO  
  
$host = "localhost";  
  
$user = "root";  
  
$senha = "";  
  
$dbname = "bd_cadastro";  
  
//CONECTA AO SERVIDOR DO MYSQL
```

```
mysql_connect($host, $user, $senha) or die("Não foi possível conectar-se com o banco de dados");
```

//SELEÇÃO DO BANCO DE DADOS

```
mysql_select_db($dbname) or die("Não foi possível conectar-se com o banco de dados");
```

//CONFIRMAÇÃO

```
if ($enviar!="") { //VERIFICANDO SE O BOTÃO FOI CLICADO E SE NÃO HÁ CAMPOS VAZIOS
```

```
if ($nome!=" " & $nascimento!=" " & $sexo!=" " & $civil!=" " & $fone!=" " & $email!=" "){
```

```
echo "<center>";
```

//INSERÇÃO NA TABELA COMANDO EM LINHA ÚNICA

```
$insere = mysql_query("INSERT INTO  
tb_dados(codigo, nome, nascimento, sexo, estado_civil, telefone, email) VALUES('", '$nome', '$nascime  
nto', '$sexo', '$civil', '$fone', '$email')") or die(mysql_error());
```

```
echo " <center> Cadastro efetuado! </center><br>";
```

```
}
```

```
else {
```

```
echo " <center> Erro! Por favor, verifique se todos os campos foram preenchidos e  
tente novamente! </center>";
```

```
}
```

```
}
```

```
?>
```

Se tudo correr bem, ao abrirmos o PHP MYAdmin basta selecionar nosso banco de dados e nossa tabela. Basta clicar no menu Visualizar, na parte superior da tela para ver os dados:

 **Visualizar**

Mostrando registros 0 - 0 (1 total. Consulta levou 0.0010 segundos)

consulta SQL:

```
SELECT *
FROM `tb_dados`
LIMIT 0 , 30
```

[Editar] [Explicar SQL] [Criar código PHP] [Visualizar]

Operações resultantes das consultas

[Visualização para impressão] [Ver impressão (com links completos)] [Exportar]

| | Mostrar . | 30 | colunas começando da 0 | | | | | | | | |
|---------|--|--------|---|------|--------------|----------|----------------------|--|--|--|--|
| no modo | horizontal | | - e repetindo cabeçalhos após 100 células | | | | | | | | |
| | codigo | nome | nascimento | sexo | estado_civil | telefone | email | | | | |
| | 1 | Rafael | 1/10/79 | masc | Casado | xxx xxx | profralph@uol.com.br | | | | |
| ↑ | Marcartodos / Desmarcartodos Com marcados: | | | | | | | | | | |

LISTAR TODOS OS DADOS (CONSULTA GERAL)

Apenas para exercitar faremos um formulário que mostra todos os dados cadastrados no banco de dados e o chamaremos de **listar.php**:

<center>
www.profralph.com.br

```

<?php
echo"<h1>LISTA DE REGISTROS</h1>";
$nome=$_POST['txtnome'];
$b=$_POST['bt'];
//CONECTAR
$db = mysql_connect("localhost","root","");
$dados = mysql_select_db("bd_cadastro",$db);
//LISTAR TODOS OS REGISTROS
$sql = mysql_query("SELECT * FROM tb_dados");
echo"<table border=0 cellspacing=1 cellpadding=2 bgcolor=#DBDBDB><tr><td
colspan=2><center>Cadastrados:</center></td><tr>";
echo "<center>";
while($linha = mysql_fetch_array($sql)){

```

```
$codigo=$linha['codigo'];

$nome=$linha["nome"];

$nascimento=$linha["nascimento"];

$sexo=$linha["sexo"];

$civil=$linha["estado_civil"];

$fone=$linha["telefone"];

$email=$linha["email"];

//IMPRIME O VETOR

echo "<td bgcolor=#66cccc>CÓDIGO:</td><td bgcolor=#66cccc>
$codigo</td><tr>";

echo "<td>NOME: </td><td>$nome</td><tr>";

echo "<td>NASCIMENTO: </td><td>$nascimento<td><tr>";

echo "<td>SEXO: </td><td>$sexo<td><tr>";

echo "<td>ESTADO CIVIL:</td><td> $civil<td><tr>";

echo "<td>TELEFONE:</td><td> $fone<td><tr>";

echo "<td>E-MAIL: </td><td>$email<td><tr>";

}

?>

</tr></table>

<br>

</center>
```

CONSULTA (BUSCA)

Criaremos um formulário para buscar registros em nosso banco de dados a partir de uma busca feita pelo usuário. Utilizaremos novamente o NVU e criaremos o formulário a seguir, chamado **consulta.php**.

FORMULÁRIO DE CONSULTA	
Nome	<input type="text"/>
<input type="button" value="Fazer a busca"/>	

<?php //RECEBEMOS OS DADOS DO FORM

\$nome=\$_POST['txtnome'];

\$enviar=\$_POST['btnconsulta'];

//DADOS DA CONEXÃO

\$host = "localhost";

\$user = "root";

\$senha = "";

\$dbname = "bd_cadastro";

//CONECTA AO SERVIDOR DO MYSQL

mysql_connect(\$host, \$user, \$senha) or die("Não foi possível conectar-se com o banco de dados");

//SELECCIONA O BANCO DE DADOS

mysql_select_db(\$dbname) or die("Não foi possível conectar-se com o banco de dados");

//CONFIRMAÇÃO

if (\$enviar!="") {

if (\$nome!="") {

echo "<center>";

//BUSCA O NOME PESQUISADO NA TABELA OU INDICA O ERRO

\$consulta = mysql_query("SELECT * FROM tb_dados WHERE nome ='\$nome'") or die(mysql_error());

//SE ENCONTRAR, CRIA UM VETOR E CARREGA OS VALORES DOS CAMPOS DA TABELA

While (\$linha=mysql_fetch_array(\$consulta)){

\$codigo=\$linha['codigo'];

\$nome=\$linha["nome"];

\$nascimento=\$linha["nascimento"];

\$sexo=\$linha["sexo"];

\$civil=\$linha["estado_civil"];

\$fone=\$linha["telefone"];

\$email=\$linha["email"];

//IMPRIME O VETOR

```

echo "<td bgcolor=#66cccc>CÓDIGO:</td><td bgcolor=#66cccc>
$codigo</td><tr>";

echo "<td>NOME:</td><td>$nome</td><tr>";

echo "<td>NASCIMENTO:</td><td>$nascimento<td><tr>";

echo "<td>SEXO:</td><td>$sexo<td><tr>";

echo "<td>ESTADO CIVIL:</td><td> $civil<td><tr>";

echo "<td>TELEFONE:</td><td> $fone<td><tr>";

echo "<td>E-MAIL:</td><td>$email<td><tr>";;

}

}

else {

    echo "<center>Erro! Por favor, verifique se o campo nome foi preenchido e tente
novamente!</center>";

}

?>
```

EXCLUSÃO (APAGAR REGISTRO)

Finalmente o formulário de exclusão (**exclui.php**). Faremos uma página que mostrará todos os dados cadastrados para que possamos decidir qual excluir e usaremos aqui um método de conexão diferente...

EXCLUSÃO DE REGISTRO

| Nome a ser apagado | | Confirmar |
|--------------------|--------------------------|-----------|
| DADOS CADASTRADOS | | |
| Cadastrado: | | |
| CÓDIGO | 16 | |
| NOME: | Rafael | |
| NASCIMENTO: | 17/10/79 | |
| SEXO: | masculino | |
| ESTADO CIVIL: | Casado | |
| TELEFONE: | xxxx xxxx | |
| E-MAIL: | profalph@qualquer.com.br | |

```
<center>

<form name=apaga action=exclui.php method=POST>

<?php

echo"<h1>EXCLUSÃO DE REGISTRO</h1>";

echo"<table border=0 cellspacing=1 cellpadding=2 bgcolor=#66cccc><tr><td>Nome a ser
apagado</td><td><input type=text name=txtnome size=30></td><td><input type=submit name=bt
value=Confirma></td></tr></table>";

$nomedigitado=$_POST['txtnome'];

$bt=$_POST['bt'];

//CONECTAR

$db = mysql_connect("localhost","root","");
$dados = mysql_select_db("bd_cadastro",$db);

//LISTAR TODOS OS REGISTROS
$sql = mysql_query("SELECT * FROM tb_dados");

echo"<table border=0 cellspacing=1 cellpadding=2 bgcolor=#DBDBDB><tr><td
colspan=2><center>Cadastrados:</center></td><tr>";

echo "<center>";

echo "DADOS CADASTRADOS:<br>";

while($linha = mysql_fetch_array($sql)){

    $codigo=$linha['codigo'];

    $nome=$linha["nome"];

    $nascimento=$linha["nascimento"];

    $sexo=$linha["sexo"];

    $civil=$linha["estado_civil"];

    $fone=$linha["telefone"];

    $email=$linha["email"];



//IMPRIME O VETOR

echo "<td bgcolor=#66cccc>CÓDIGO:</td><td      bgcolor=#66cccc>
$codigo</td><tr>";
```

```
echo "<td>NOME:</td><td>$nome</td><tr>";  
echo "<td>NASCIMENTO:</td><td>$nascimento<td><tr>";  
echo "<td>SEXO:</td><td>$sexo<td><tr>";  
echo "<td>ESTADO CIVIL:</td><td> $civil<td><tr>";  
echo "<td>TELEFONE:</td><td> $fone<td><tr>";  
echo "<td>E-MAIL:</td><td>$email<td><tr>";  
}  
  
if($bt!="){  
  
//APAGAR O NOME ESCOLHIDO  
  
$sql = mysql_query ("DELETE FROM tb_dados WHERE nome ='$nomedigitado'")or  
die(mysql_error());  
  
echo"<script language=javascript>  
alert('Participante cadastrado com sucesso!!');  
</script>";  
  
//ATUALIZAR A TELA  
  
echo"<meta http-equiv='refresh' content='0' />";  
  
}  
  
?  
  
</tr></table><br>  
</form>  
</center>
```

SORTEAR INFORMAÇÕES DO BANCO DE DADOS

Por uma necessidade qualquer podemos fazer com que o PHP sorteie aleatoriamente do banco de dados informações para serem mostradas. Esta técnica é muito utilizada em sites em que o banner fica mundo sempre que o site é atualizado ou um link é clicado.

Vamos criar um script chamado **sorteia.php**:

SORTEIO DE REGISTROS

[Clique para sortear](#)

O código será:

```
<center>

<h1>SORTEIO DE REGISTROS</h1>

<form action="sorteia.php" method="POST">

<input type="submit" name="bt" value="Clique para sortear">

</form>

<?php

$nome=$_POST['txtnome'];

$bt=$_POST['bt'];

//CONECTAR

$db = mysql_connect("localhost","root","");
dados = mysql_select_db("bd_cadastro",$db);

//LISTAR TODOS OS REGISTROS

$sql = mysql_query("SELECT * FROM tb_dados");

echo"<table border=0 cellspacing=1 cellpadding=2 bgcolor=#DBDBDB><tr><td
colspan=2><center>Cadastrados:</center></td><tr>";

echo "<center>";

if($bt!="){

//MOSTRAREMOS 2 REGISTROS SORTEADOS, MAS PARA MOSTRAR MAIS, BASTA ALTERAR O
NÚMERO DE LIMIT

$sql = mysql_query("SELECT * FROM tb_dados ORDER BY rand() LIMIT 2");

while($linha = mysql_fetch_array($sql)){

$codigo=$linha['codigo'];

$nome=$linha["nome"];

$nascimento=$linha["nascimento"];

$sexo=$linha["sexo"];
```

```
$civil=$linha["estado_civil"];  
  
$fone=$linha["telefone"];  
  
$email=$linha["email"];  
  
//IMPRIME O VETOR  
  
echo "<td bgcolor=#66cccc>CÓDIGO:</td><td bgcolor=#66cccc>  
$codigo</td><tr>";  
  
echo "<td>NOME: </td><td>$nome</td><tr>";  
  
echo "<td>NASCIMENTO: </td><td>$nascimento<td><tr>";  
  
echo "<td>SEXO: </td><td>$sexo<td><tr>";  
  
echo "<td>ESTADO CIVIL:</td><td> $civil<td><tr>";  
  
echo "<td>TELEFONE:</td><td> $fone<td><tr>";  
  
echo "<td>E-MAIL: </td><td>$email<td><tr>";  
  
}  
  
}  
  
?  
</tr></table><br>  
</center>
```

Testando o código, percebe-se que ao clicar no botão serão sorteados e exibidos dois registros.

Agora que terminamos todas as operações com Banco de Dados, podemos criar uma página chamada **index.html** com link para todas as demais e brincar com todas as páginas.

LOGIN COM SESSÃO

Quando queremos impedir o acesso de pessoas indesejadas a determinadas páginas de nosso site podemos utilizar o recurso de login por autenticação de usuário, ou seja, exigir um cadastro e autorizar o acesso a pessoas cadastradas por meio de senha.

Um login simples pode ser feito com simples comandos, mas pode ser burlado através da barra de endereços do navegador, mas trabalhando com sessões, este problema é resolvido.

Primeiro criaremos uma pasta em c:\wamp\www com o nome **login_sessao** e começaremos com uma página chamada **index.html** com **form action='autentica.php'** para o acesso:

ACESSO À PÁGINA RESTRITA

| | |
|--|--------------------------|
| Apelido: | <input type="text"/> |
| Senha: | <input type="password"/> |
| <input type="button" value="Logar"/> | |
| Ainda não é cadastrado? Clique Aqui | |

Esta página estará através do link **Clique Aqui** ligada a outra página chamada **cadastro.html** com **form action='caduser.php'**:

CADASTRO DE USUÁRIOS

| | |
|---|--------------------------|
| Nome | <input type="text"/> |
| Apelido | <input type="text"/> |
| E-mail | <input type="text"/> |
| Senha | <input type="password"/> |
| Confirmar senha | <input type="password"/> |
| <input type="button" value="Confirmar cadastro"/> | |

Agora criaremos os scripts PHP responsáveis por toda a operação. Usaremos uma forma diferente de trabalhar. Vamos criar um script responsável por cada operação, sendo um para conexão (**conn.php** usado para conectar com o Banco de Dados), um para cadastrar (**caduser.php**), um para efetuar o login (**autentica.php**), um para efetuar o logoff (**sair.php**), nossa página restrita (**result.php**) e um para verificar se a pessoa realmente logou e autorizar o acesso a nossa página restrita (**verifica.php**) começaremos com a conexão.

CONEXÃO:

Abra o Notepad++, digite o código a seguir e salve como **conn.php**:

```
<?php

$host="localhost";
$banco="bd_cadastro";
$user="root";
$senha="";
$conn = @mysql_connect($host,$user,$senha) or die ("Erro ao conectar-se");
$db = @mysql_select_db($banco,$conn) or die ("Erro ao abrir o Banco");

?>
```

CADASTRO:

Abra o Notepad++, digite o código a seguir e salve como **caduser.php**:

```
<?php

require_once("conn.php"); //USA A CONEXÃO conn.php

$nome=$_POST[txtnome];

$apelido=$_POST[txtapelido];

$apelido=addslashes($apelido);

$senhauser=$_POST[txtsenha];

$senha=$_POST[txtconfsenha];

$senha=md5($senha); //CRIPTOGRAFA A SENHA

$senhauser=md5($senhauser); //CRIPTOGRAFA A SENHA

$email=$_POST[txtemail];

$cadastrar=$_POST[btncadastrar];

if ($cadastrar!="){

    if ($senhauser==$senha){

        //INSERÇÃO NA TABELA

        $insere = mysql_query("INSERT INTO

tb_usuario(id_usuario, nome, apelido, senha, email)VALUES('','$nome','$apelido','$senhauser','$email')

")or die(mysql_error());

        echo "<script> alert('Cadastro Salvo é só fazer o login!')</script>";

        echo "<script Language='javascript'>

window.location.href='index.html'</script>"; //OU Header("Location: index.html");

    }

    else{

        echo '<center><h2>As senhas digitadas nas caixas de texto não são iguais!

Volte e tente novamente!';

    }

}

?>
```

AUTENTICAÇÃO:

Abra o Notepad++, digite o código a seguir e salve como **autentica.php**:

```
<?php
```

```
require_once("conn.php");

$apelido=$_POST[txtapelido];

$apelido=addslashes($apelido);

$senhauser=$_POST[txtsenha];

$senhauser=md5($senhauser);

$entrar=$_POST[btnentrar];

if ($entrar!=""){

    $sql=mysql_query("SELECT id_usuario, nome FROM tb_usuario
                      WHERE apelido='".$apelido.' AND senha='".$senhauser.' ") or die
                      ("<center><h2>Senha e/ou usuário errado!</h2></center>");

    $row=mysql_num_rows($sql);

    if ($row==0){

        echo "<script> alert('Não há usuário cadastrado!Faça o seu
cadastro!')</script>";

        echo"<script Language='javascript'
window.location.href='index.html'</script>"; //OU Header("Location: index.html");

    }else{

        $id=mysql_result($sql, 0, "id_usuario");

        $nome=mysql_result($sql, 0, "nome");

        session_start();

        $_SESSION[iduser]=$id;

        $_SESSION[nomeuser]=$nome;

        echo"<script Language='javascript'
window.location.href='result.php'</script>"; //OU Header("Location: result.html");

    }

}
```

```
?>
```

OBS: **result.php** é a minha página restrita deste exemplo. Coloque o nome da sua.

PÁGINA RESTRITA:

Esta é a página que só aparecerá se o usuário estiver cadastrado e efetuar o logon. Pode ter qualquer conteúdo e ser de qualquer tipo (HTML, htm, php, etc).

SÓ NÃO SE ESQUEÇA de acrescentar **require_once("verifica.php");** para que possa ser feita a verificação se a autenticação foi feita e a sessão iniciada e ainda colocar um link para **sair.php** para ser feito o logoff.

VERIFICAÇÃO DE AUTENTICAÇÃO:

Abra o Notepad++, digite o código a seguir e salve como **verifica.php**:

```
<?php  
  
    session_start();  
  
    if((!isset($_SESSION[iduser])) AND (!isset($_SESSION[nomeuser])))  
  
    {  
  
        echo"<script Language='javascript'>  
window.location.href='index.html'</script>"; //OU Header("Location:index.html");  
  
    }  
  
?>
```

EFETUAR LOGOFF:

Abra o Notepad++, digite o código a seguir e salve como **sair.php**:

```
<?php  
  
    session_start();  
  
    unset($_SESSION[iduser]);  
  
    unset($_SESSION[nomeuser]);  
  
    echo"<script Language='javascript'> window.location.href='index.html'</script>";  
//OU Header("Location: index.html");  
  
?>
```

ATIVIDADES (PHP COM BANCO DE DADOS):

1. Refaça todos os exercícios anteriores usando o NVU (com código híbrido e separado) e aplique todos os tratamentos de erros necessários. Crie bancos de dados e armazene todas as informações nestes bancos de dados.
2. Fazer uma página para cadastro de alunos que concorrerão a um prêmio e outra que fará o sorteio dos alunos e mostrará quem são os contemplados.

EXPORTAÇÃO E IMPORTAÇÃO DE BANCO DE DADOS NO PHP MYADMIN

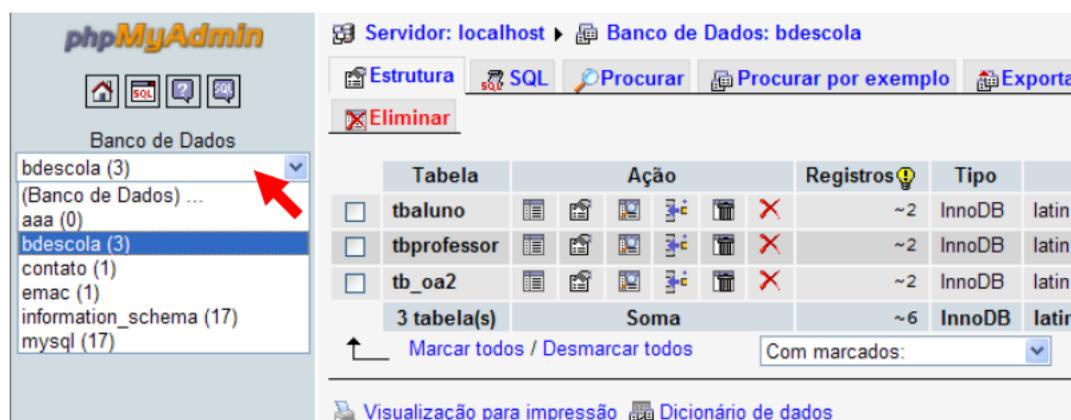
Quando precisamos em uma máquina utilizar um banco de dados com todos os registros que foi feito em outro computador (por exemplo o banco feito na sua casa ser aproveitado na escola) basta trabalhar com o recurso de exportação e importação do PHP MyAdmin.

Você exporta o banco de dados pronto, para um arquivo sql e este, por sua vez é importado no outro computador que você estiver utilizando no momento.

Nota: Ambos os computadores deverão possuir o wampserver instalado.

EXPORTANDO O BANCO DE DADOS PRONTO:

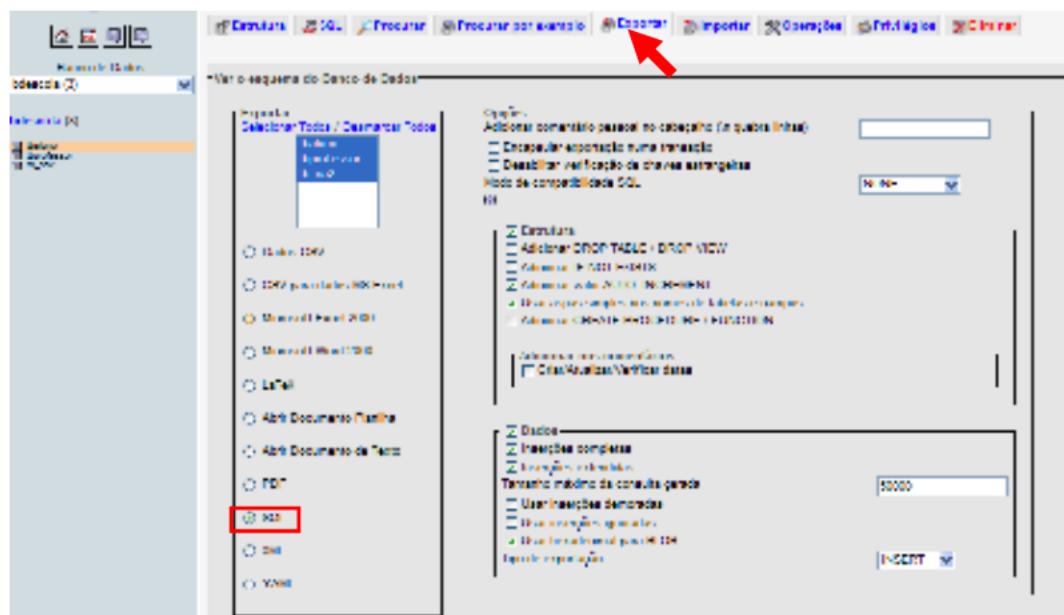
Abra o PHP MyAdmin e selecione o banco de dados que deseja exportar.



The screenshot shows the phpMyAdmin interface. On the left, there is a sidebar titled "Banco de Dados" with a dropdown menu. The "bdescola (3)" option is selected and highlighted with a blue background. A red arrow points to this selection. On the right, there is a main panel titled "Servidor: localhost > Banco de Dados: bdescola". It contains tabs for "Estrutura", "SQL", "Procurar", "Procurar por exemplo", and "Exporta". Below these tabs, there is a "Eliminar" button. The main area displays a table of databases and their details. The table has columns: Tabela, Ação, Registros, Tipo, and Collation. It lists three tables: "tblaluno", "tbprofessor", and "tb_oa2", all of which have "InnoDB" as their type and "latin1" as their collation. The total number of rows is approximately 6. At the bottom of the table area, there are buttons for "Visualização para impressão" and "Dicionário de dados".

Caso ache necessário, clique em uma das tabelas do Banco de Dados e em Visualizar para ver os dados já cadastrados.

Com o Banco de Dados selecionado, clique no botão Exportar, na parte superior da tela.



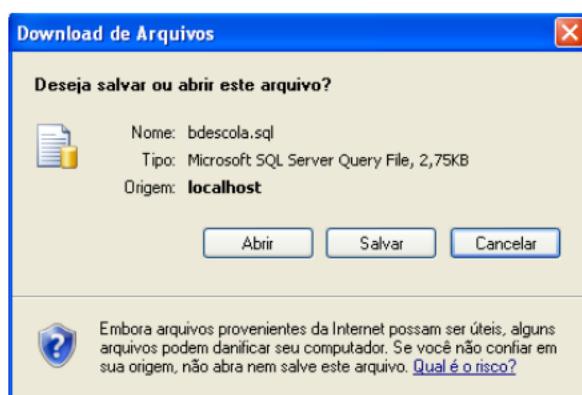
Nesta janela você dever ter certeza de que todas as tabelas estejam selecionadas. No canto esquerdo, selecione a opção SQL.

Repare agora na parte inferior da tela:



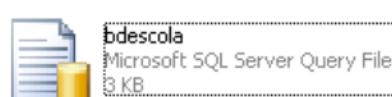
Selecione a opção Enviado para que o arquivo SQL seja gerado. Caso você se esqueça de marcar a esta opção, será gerado o código, mas na janela do MyAdmin e este não será salvo.

Clique no botão Executar.



Na caixa de diálogo, selecione a opção Salvar e grave o arquivo onde desejar.

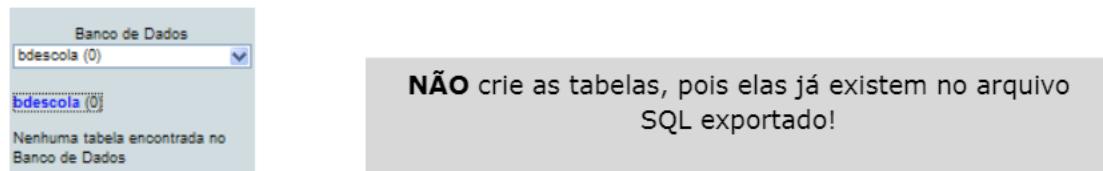
Se tudo correr bem, será gerado um arquivo como na imagem a seguir:



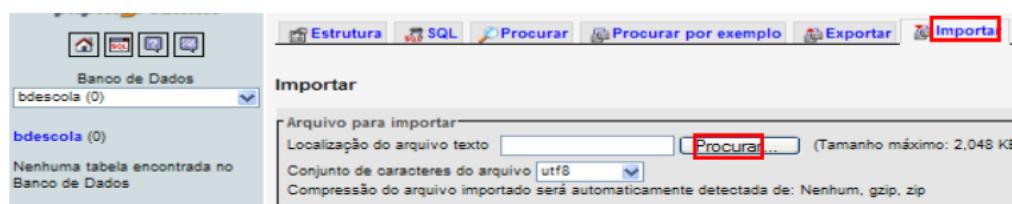
Este arquivo contém a estrutura das tabelas e o conteúdo armazenado nas mesmas e será utilizado no processo de importação.

IMPORTANDO O BANCO DE DADOS EM OUTRA MÁQUINA:

No outro computador, abra o MyAdmin e crie um banco de dados com o mesmo nome do que foi exportado.



Clique no botão Importar, na parte superior da tela e selecione o arquivo com o botão Procurar:



Para finalizar, clique no botão Executar, no canto inferior direito.



Automaticamente o MyAdmin fará a atualização e as tabelas já poderão ser vistas no menu lateral esquerdo:



Clicando nas tabelas, você poderá ver os dados nas mesmas através da opção Visualizar.

| | cod_professor | nome | disc | sigla | sem | ano | turma | curso | usuario |
|--------------------------|---------------|----------------|--|-------|-----|------|----------|-------------|---------|
| <input type="checkbox"/> | 1 | D?bora Vicente | Opera??o de Softwares e Aplicativos II | OAI1 | 2 | 2009 | 2 INF T | INFORMATICA | debora |
| <input type="checkbox"/> | 17 | Rafael | Aplicativos para Internet | AI | | 2009 | Terceiro | Info | rafa |

Importação efetuada com sucesso!