

# CURSO PHP/MySQL

## **Módulo I**

Características da linguagem PHP – 2

Preparando o ambiente de desenvolvimento com Xampp – 2

Acesso e configuração do Eclipse – 6

Criando o primeiro projeto PHP no Eclipse – 12

Digitando o primeiro código PHP - 13

## **Módulo II**

Criando o primeiro projeto PHP – 7

Conhecendo o Eclipse, criando a primeira Classe- 8

Digitando, salvando e executando o primeiro código – 9

Tags, comentários – 14

Variáveis, tipos de dados, Strings, comando Settype – 15

Operadores – 16

Controle de fluxo If Else, Switch, While e For – 17

Variáveis de instância e variáveis locais, mais código – 15

Operadores – 16

Controle de Fluxo (IF else, While, For), mais códigos – 17

## **Módulo III**

Banco de Dados MySQL com PHP – 20

Criando e acessando banco de dados através do phpMyAdmin – 21

Conectando o bando de dados – 22

## **Módulo IV**

Sistema usuário e senha com PHP, MySQL e HTML – 23

Criando um novo projeto PHP – 25

Acessando um arquivo PHP pelo navegador - 27

Criando um arquivo HTML - 27

Finalizando o sistema de usuário e senha com verificação MySQL - 29

## Módulo I - Características e preparação do ambiente de desenvolvimento

O PHP é uma linguagem de programação de script, Orientada a Objetos, open source e seu foco é programação web que pode ser embutida dentro de um código HTML.

### :: Principais Características:

- É multiplataforma, tendo suporte aos sistemas Operacionais mais utilizados no mercado;
- Tem suporte a um grande número de bancos de dados como: dBase, Interbase, mSQL, MySQL, Oracle, Sybase, PostgreSQL e vários outros.
- Portabilidade, independe de plataforma, ou seja, escreva uma vez, rode em qualquer lugar.
- Seu código é livre, não é preciso pagar por sua utilização e pode ser alterado pelo usuário na medida da necessidade de cada usuário
- Estruturado e orientação a objetos
- Open-source

### :: Preparando o ambiente de desenvolvimento com XAMPP

O XAMPP é um pacote que já vem com Apache +PHP + MySQL pronto pra ser usado e simular um servidor web local. Onde Apache é o servidor, MySQL é o banco de dados e PHP é a linguagem de programação. O XAMPP Possui suporte a outras linguagens de programação além do PHP, como Perl por exemplo.

Atualmente o XAMPP está disponível para quatro sistemas operacionais: Windows, Linux, Mac OS X e Solaris . Neste curso iremos utilizar a versão para Windows.

Vamos à instalação.

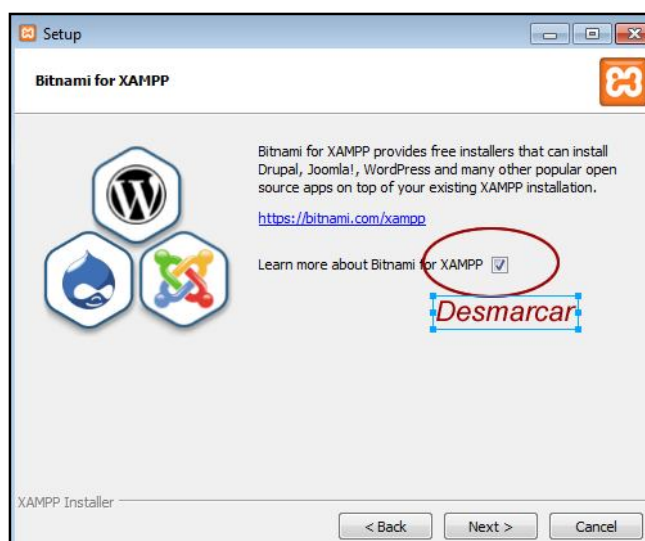
1 – Baixar o XAMPP no endereço [https://www.apachefriends.org/pt\\_br/index.html](https://www.apachefriends.org/pt_br/index.html) e instalar normalmente.



Após baixar e executar o arquivo você terá a seguinte tela:



Pode ir clicando em Next normalmente até chegar na seguinte tela e desmarcar esta opção :



Next até chegar em:



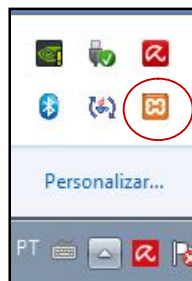
Ao término, basta clicar em Finish e escolher o inglês como idioma (não tem português) e clicar em Save.



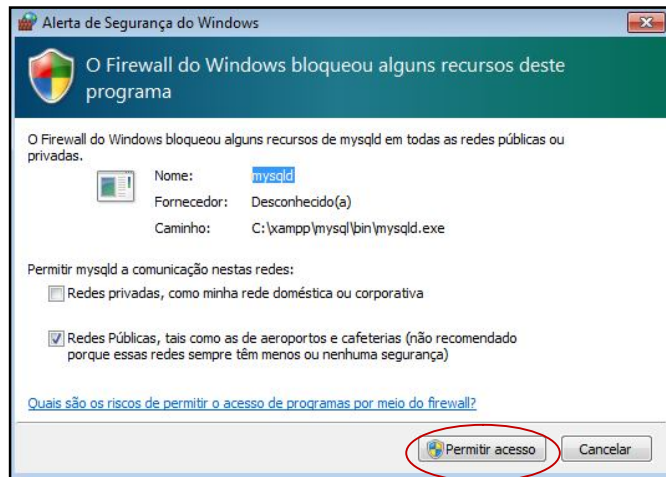
Feito isso você terá acesso ao painel de controle do XAMPP onde você irá iniciar (start) o Apache e o MySQL como na figura abaixo.



Assim que você iniciar os dois servidores, pode fechar o painel de controle do XAMPP que ele ficará minimizado na área de notificação do Windows.



Pode ser que seja necessário liberar o Apache no Firewall do Windows

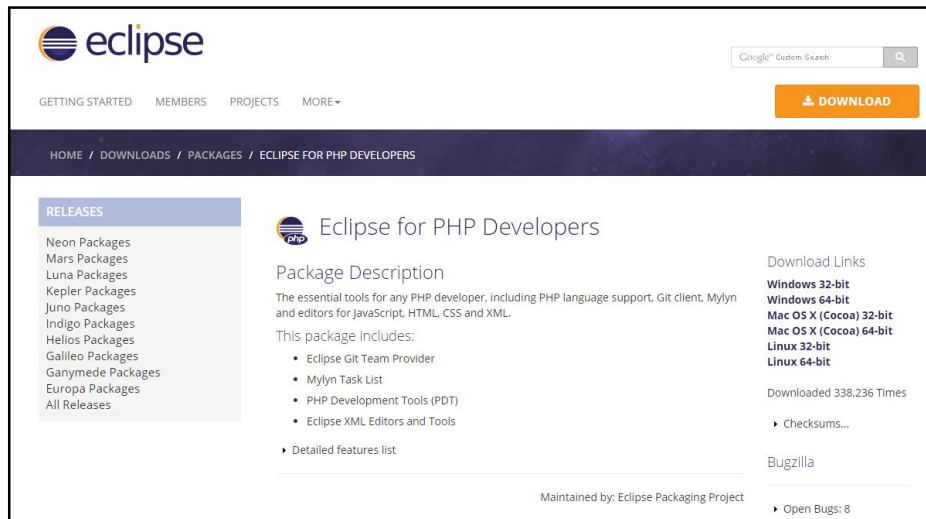


Após a instalação do XAMPP e o start dos serviços Apache e MySQL, vamos testar pra ver se está tudo ok. Acesse o endereço de localhost no seu navegador. No meu caso estou utilizando o Google Chrome, mas qualquer navegador serve. Entre no navegador e acesse o endereço <http://127.0.0.1/> ou <http://localhost> e, se estiver tudo ok, você verá uma tela de boas vindas do XAMPP.

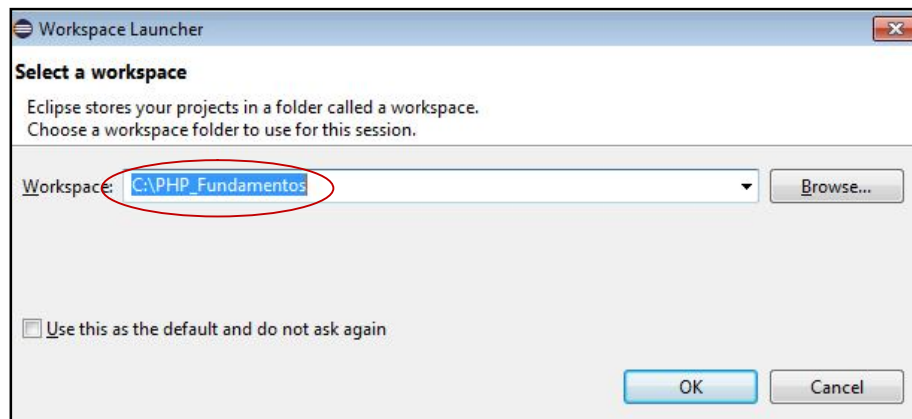


Agora que já instalamos e testamos o XAMPP, vamos baixar e instalar um editor PHP para facilitar a digitação dos códigos. Você poderá usar um editor simples como o Bloco de Notas por exemplo, mas neste curso eu irei adotar o **Eclipse PHP** que é um plug in para desenvolvimento PHP, ele possui muito mais recursos de programação.

Faça o download do Eclipse PHP no endereço <http://www.eclipse.org/downloads/packages/eclipse-php-developers/lunar>. No meu caso eu utilizei a versão 32-bit.



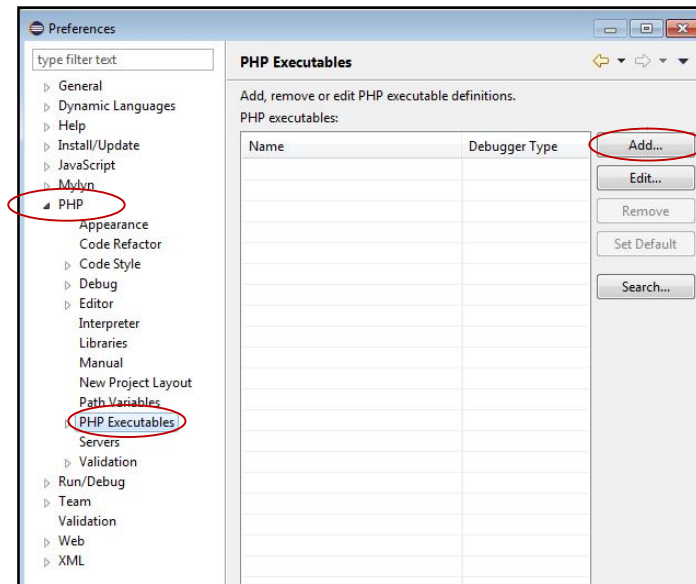
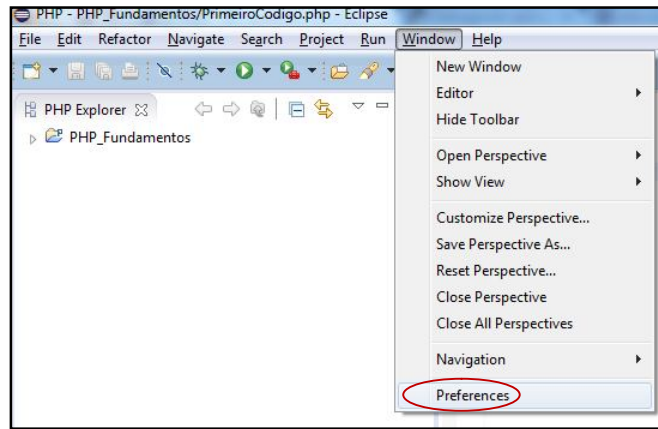
Após baixar e executar o Eclipse PHP pela primeira vez ele irá perguntar qual o seu Workspace, que nada mais é do que a pasta onde você irá armazenar seus códigos. No meu caso eu criei uma pasta chamada PHP\_Fundamentos. Escolha uma pasta e clique OK.



Feche a tela de boas vindas.

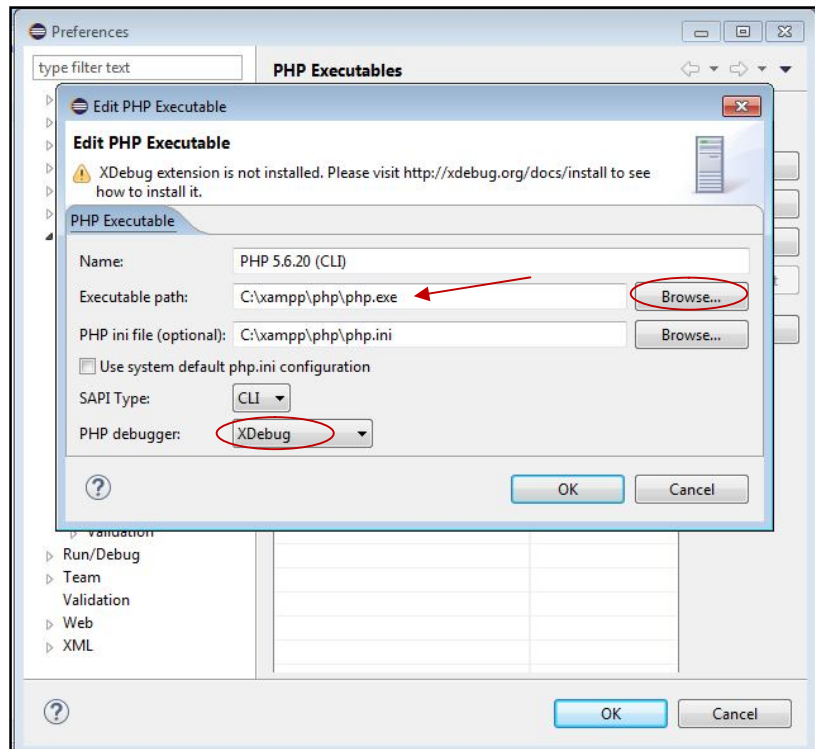


Agora vamos configurar o Eclipse PHP, vamos indicar onde se encontra o php.exe na pasta do XAMPP. Clique Window, Preferences, PHP, PHP Executable, Add, Browse e encontre o arquivo **php.exe** que deve estar na pasta **c:\xampp\php\**.

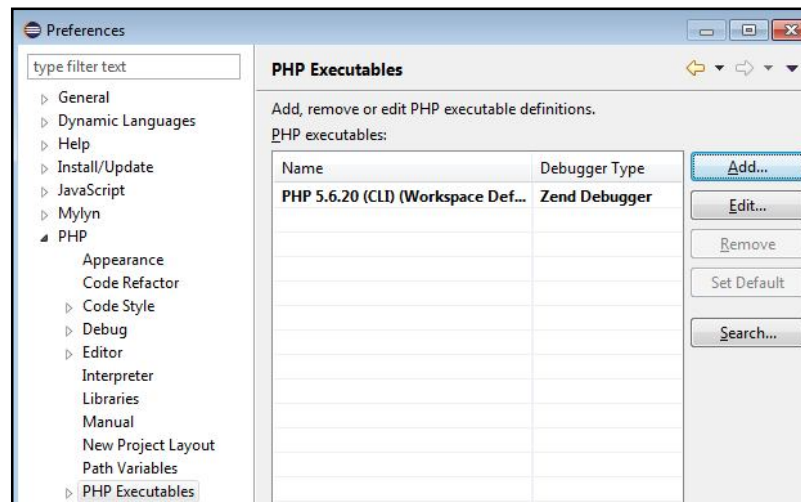


Clique em Add encontre o arquivo php.exe na pasta c:\xampp.

Altere o PHP debugger para XDebug



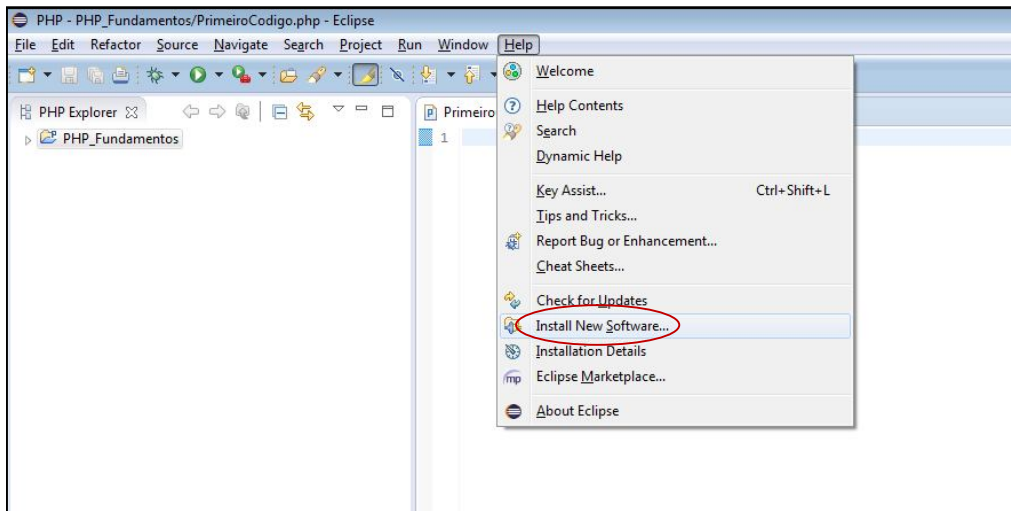
Clique Ok.



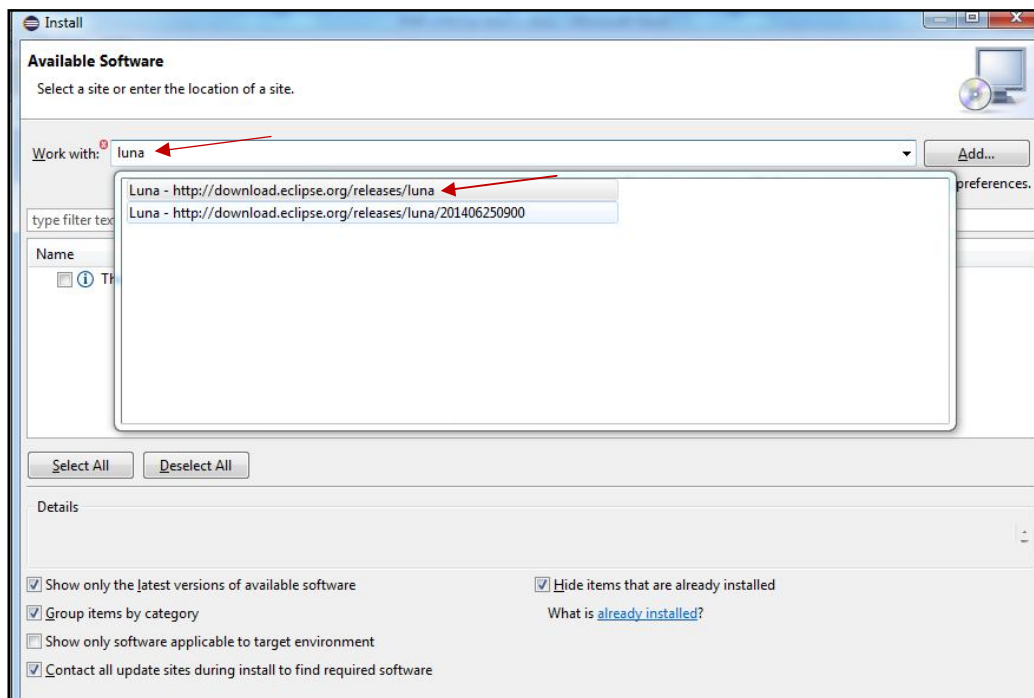
Agora iremos instalar o plug in PHP para Eclipse. Estamos quase terminando.

Acesse o menu Help e clique em Install new software.

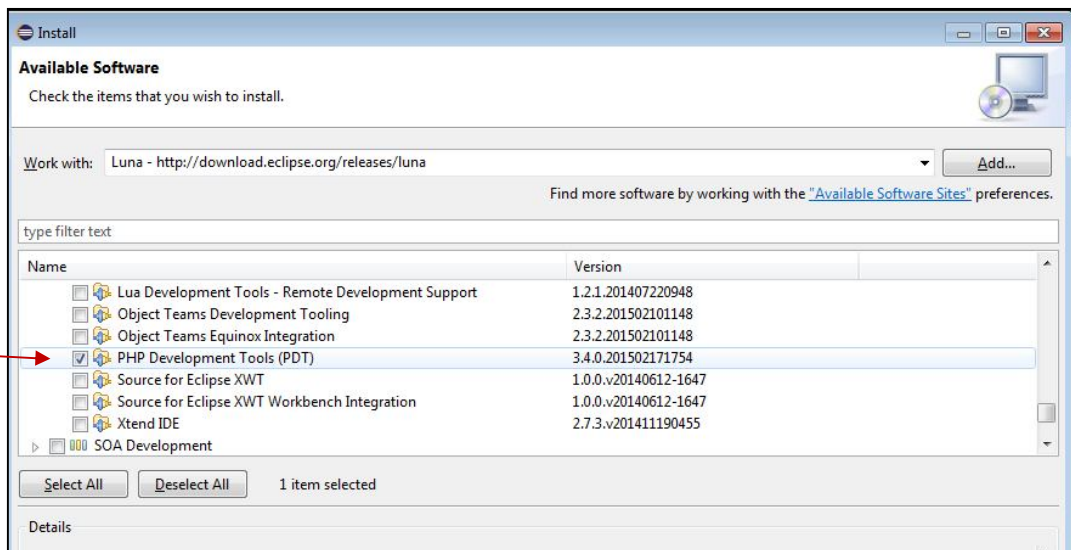
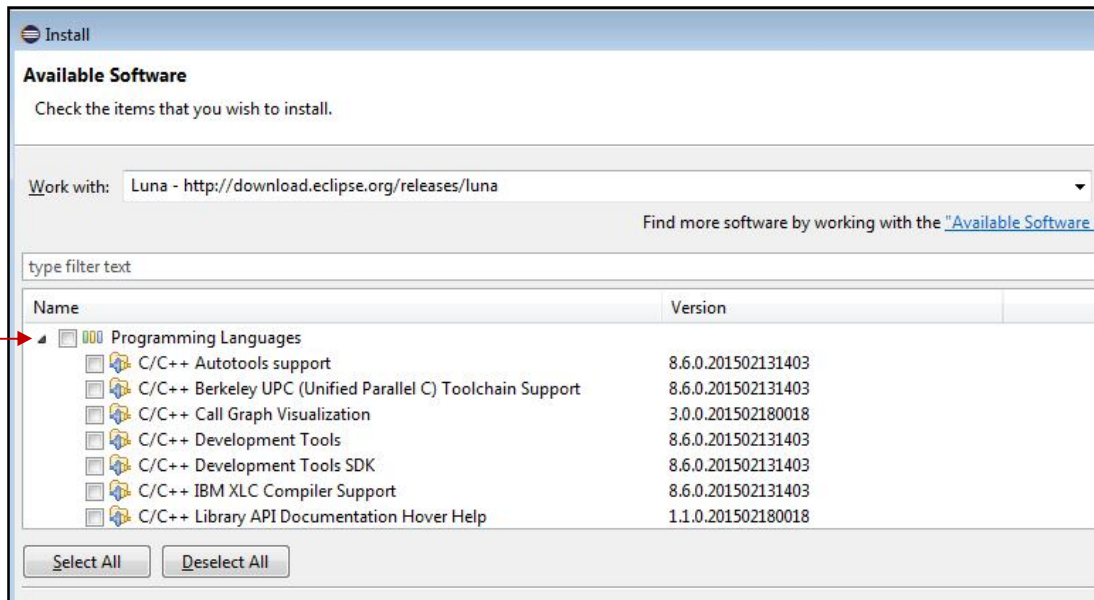




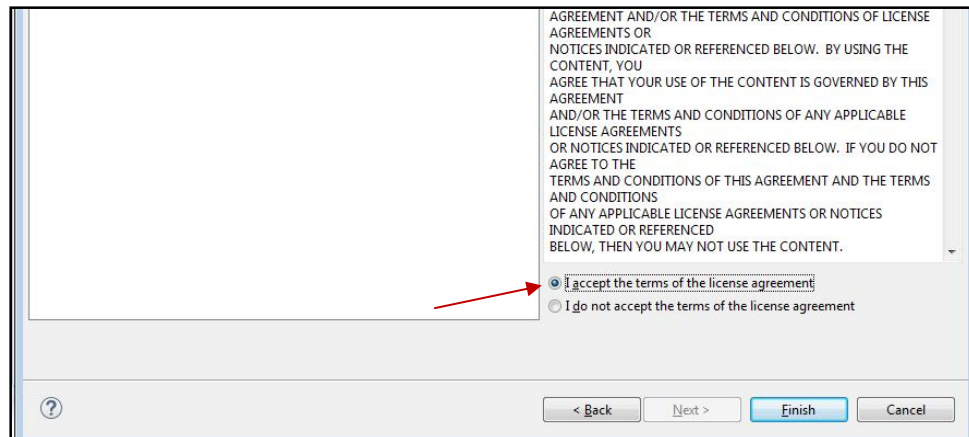
Em **Work with** digite **luna** (que é a versão do eclipse que estamos utilizando) e escolha a opção como mostrado na figura abaixo.



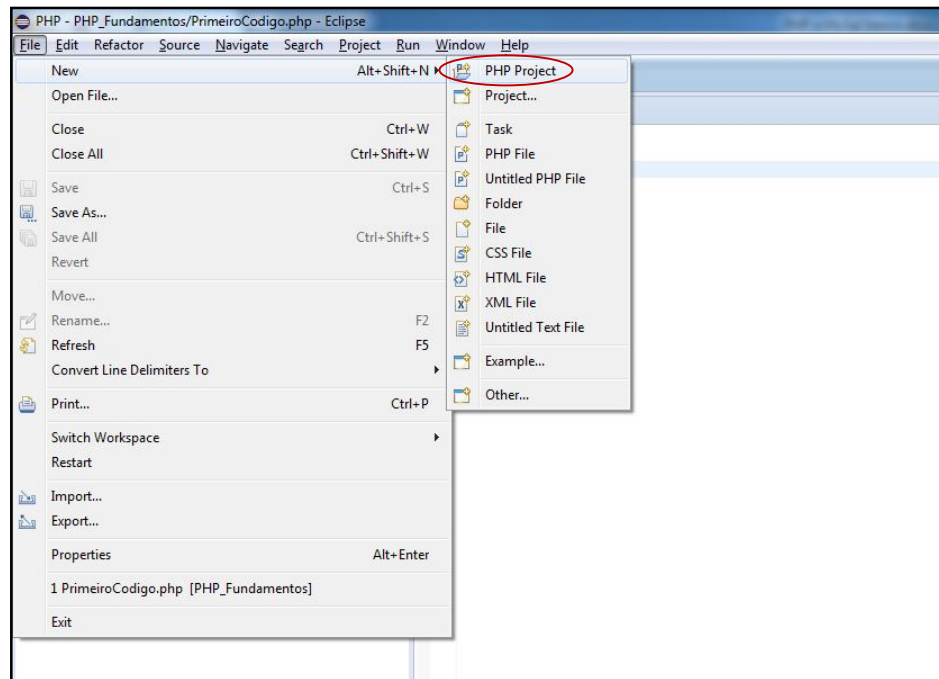
Agora clique na setinha da opção Programming Languages e marque a **opção PHP Development Tools (PDT)**



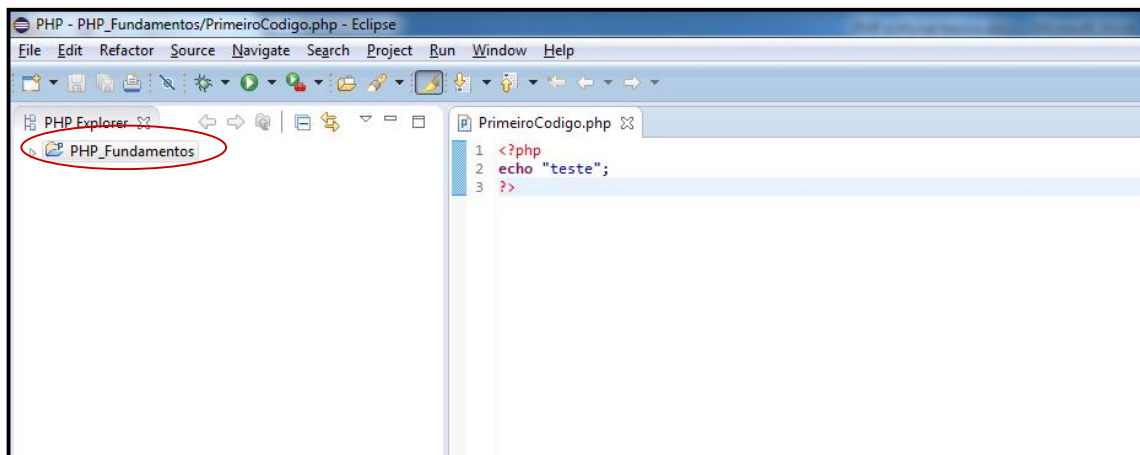
Clique em Next, aceite os termos de licença e clique em **Finish**. O Eclipse vai se reiniciar e acabou, nosso ambiente de desenvolvimento finalmente está pronto para ser usado.



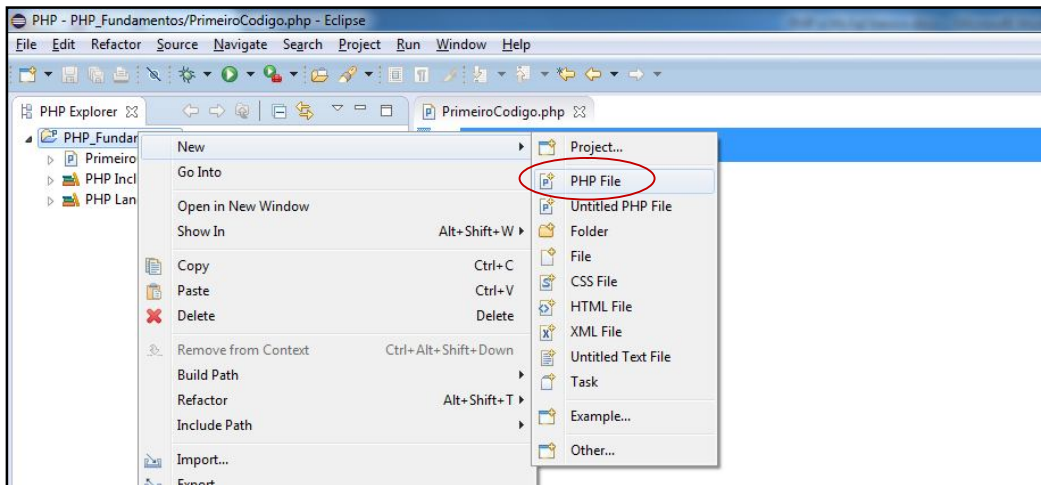
Vamos fazer um teste já digitando o nosso primeiro código PHP. Primeiro temos que criar um projeto PHP: clique em File, New, PHP Project.



Em Project Name digite o nome do projeto que no meu caso é o mesmo nome do workspace, PHP\_Fundamentos. Clique em Finish. Você terá o seu projeto criado como a figura abaixo.

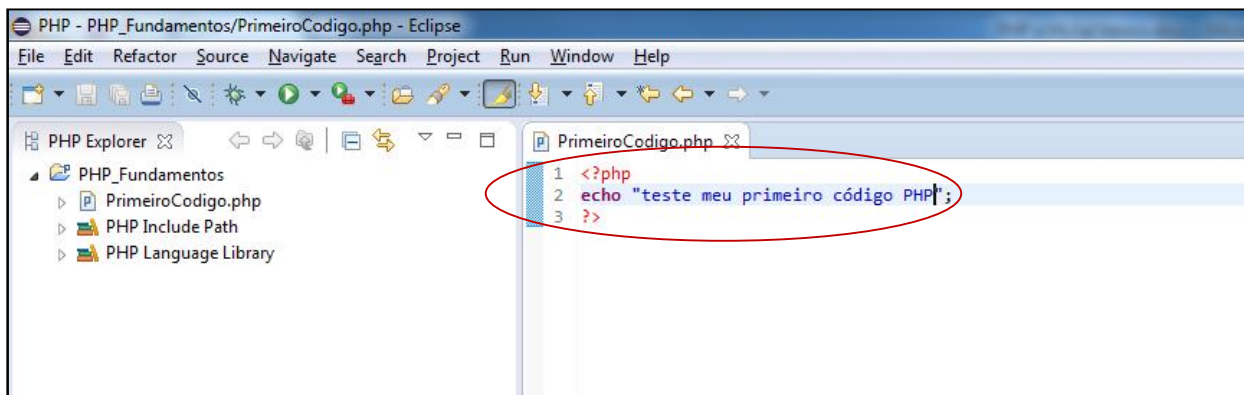


Agora vamos criar o nosso arquivo PHP. Clique com o botão direito do mouse no projeto criado (PHP\_Fundamentos), New, PHP File




Em File Name digite o nome do arquivo, **PrimeiroCodigo.php** e clique em Finish.

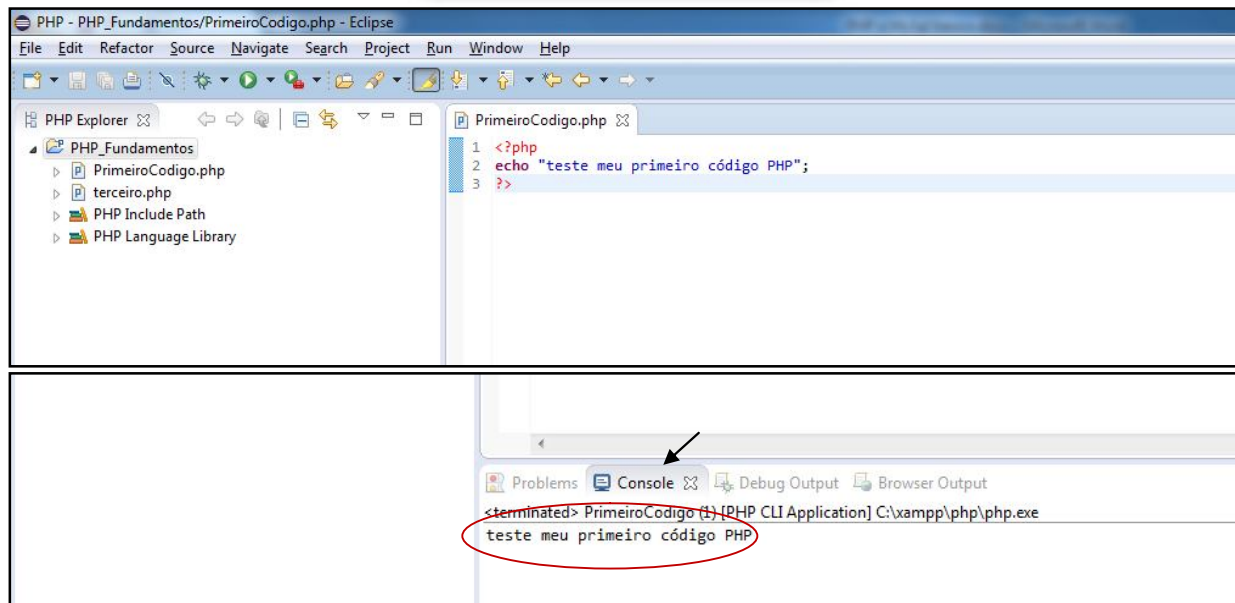
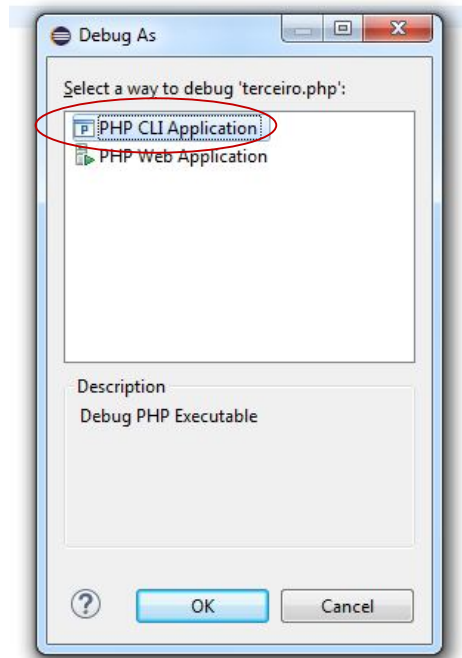
Digite o código conforme figura abaixo. Perceba que o texto fica entre aspas.



Estamos diante do nosso primeiro comando PHP que é o “**echo**”. Este comando serve para exibir um texto na tela. Também devemos notar que todo o código PHP fica entre <?php ... ?>.

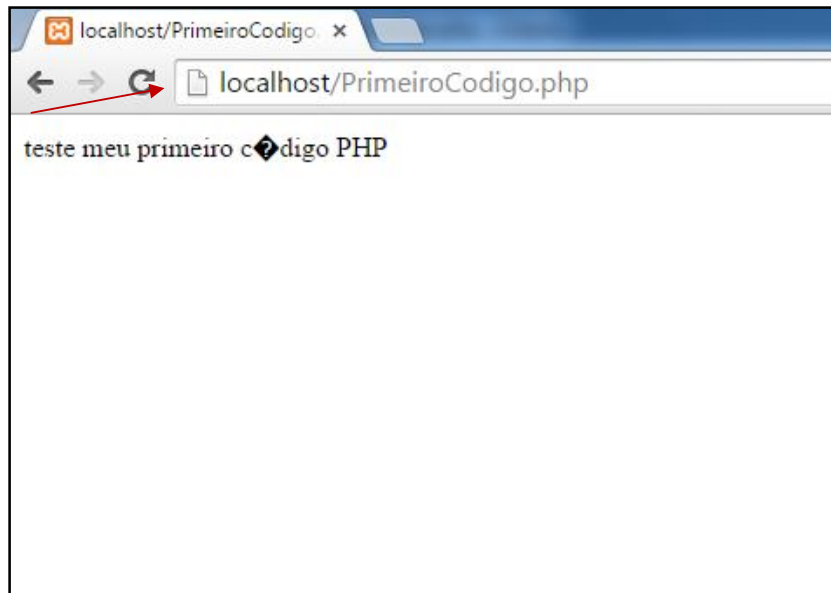
Tecle CTRL + S para salvar as alterações.

Tecle F11 ou clique em  para executar o código. Ao executar pela primeira vez este código você verá a seguinte tela. Escolha a opção de Debug PHP CLI Application (isso é para executarmos os códigos dentro do próprio Eclipse), clique OK e você verá o código executado na guia **console**.



Detalhe: tudo isso que fizemos foi para podermos programar em PHP na IDE Elipse e assim podermos executar/testar nossos códigos no próprio Eclipse. Mas vamos testar este código no navegador, afinal a nossa intenção aqui é justamente programar para Web.

Vamos executar este nosso primeiro código no navegador. Primeiro entre em **C:\PHP\_Fundamentos\PHP\_Fundamentos**, copie o arquivo PrimeiroCodigo.php e cole em **C:\xampp\htdocs**. Agora entre no navegador e acesse o endereço como na figura abaixo e tecle Enter.



Pronto, agora você tem uma página web em PHP. Então é isso, podemos utilizar somente o Eclipse para testarmos os nossos códigos, mas podemos também fazer os testes direto no navegador colocando-os na pasta **C:\xampp\htdocs**.

## **Módulo II – Delimitadores (tags), comentários, variáveis tipos de dados, coerção**

### **:: Delimitadores ou tags**

Para o PHP ser entendido pelo navegador web ele já fica embutido no próprio código HTML e o interpretador identifica o código PHP através dos delimitadores ou tags **<?php ?>**. Ex:

```
<? php
```

```
Comandos...;
```

```
Comandos...;
```

```
?>
```

Obs: Lembrando que cada comando PHP será seguido de ponto e vírgula ao final da linha.

### **:: Comentários**

Digamos que você queira fazer uma observação em um código para lembrar alguma coisa no futuro quando você ver o tal código novamente ou fazer alguma observação em uma linha, observação esta que você não poderá esquecer na próxima vez que abrir o seu programa, nesse caso você utiliza de comentários. Esses comentários simplesmente não interferem no código, você pode digitar o que quiser.

Comentário só de uma linha é feito com duas barras //. Comentário de mais de uma linha você inicia com /\* e termina com \*/. Os comentários são simplesmente ignorados pelo compilador, não interferem no código.

## :: Variáveis

As variáveis em PHP sempre serão precedidas pelo caracter dólar \$. Ex: \$var, \$nome

Lembrando que o PHP é case sensitive, ou seja, ele diferencia maiúsculas de minúsculas. Então \$var é diferente de \$VAR.

## :: Tipos de Dados Suportados

O PHP suporta os tipos:

- Inteiro
- Ponto flutuante
- String
- Array
- Objeto

Detalhe: não é preciso declarar o tipo de dado na declaração da variável porque o PHP utiliza checagem dinâmica de variáveis, ou seja, uma variável poderá conter diferentes valores de diferentes tipos durante a execução do código, o interpretador PHP decide qual o melhor tipo de dado representa aquele valor. Mesmo assim o programador pode realizar uma conversão para um determinado tipo de dado se assim o desejar utilizando, por exemplo, o comando **settype** que veremos adiante.

## :: Strings

Strings podem ser atribuídas de duas maneiras:

- utilizando aspas simples ( ' ) – Desta maneira, o valor da variável será exatamente o texto contido entre as aspas .
- utilizando aspas duplas ( " ) – Desta maneira, qualquer variável ou caracter de escape será expandido antes de ser atribuído.

## :: Coersão

Convertendo um tipo primitivo em outro explicitamente e com o comando **settype**.

Forma explícita:

```
<?php
$i = 10;
$i = (double)$i; //agora $i é um Double
?>
```

### Comando Settype

O comando settype converte uma variável para um tipo de dado especificado pelo programador.  
*Settype(variável,novo tipo da variável).*

```
<?php
```

```
$i =10;
```

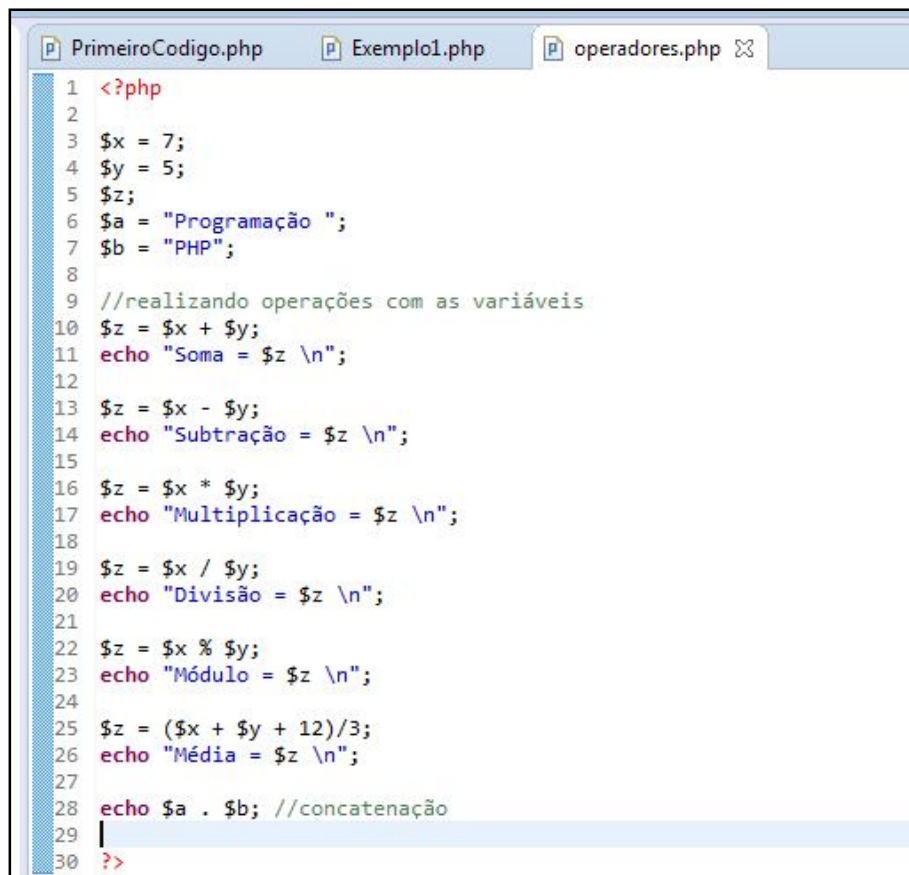
```
settype ($i, Double); //agora $i é um Double
```

```
?>
```

:: **Operadores.** Os principais são:

Soma	+	E	&&
Subtração	-	Ou	
Multiplicação	*	Igual a	==
Divisão	/	Diferente de	!=
Módulo	%	Ternário	? :
Concatenação	.	Incremento	++
Atribuição simples	=	Decremento	--
Atribuição com adição	+=		

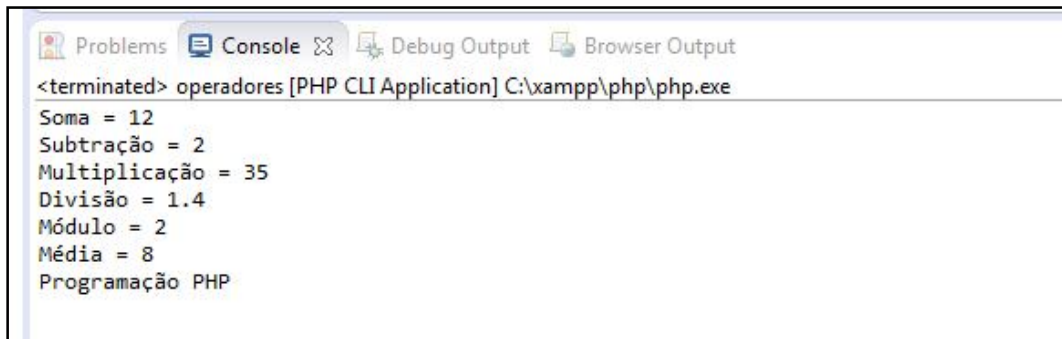
Exemplo de código utilizando operadores. Crie um arquivo chamado Operadores.php. No Eclipse PHP : File, New, PHP File.



```
1 <?php
2
3 $x = 7;
4 $y = 5;
5 $z;
6 $a = "Programação ";
7 $b = "PHP";
8
9 //realizando operações com as variáveis
10 $z = $x + $y;
11 echo "Soma = $z \n";
12
13 $z = $x - $y;
14 echo "Subtração = $z \n";
15
16 $z = $x * $y;
17 echo "Multiplicação = $z \n";
18
19 $z = $x / $y;
20 echo "Divisão = $z \n";
21
22 $z = $x % $y;
23 echo "Módulo = $z \n";
24
25 $z = ($x + $y + 12)/3;
26 echo "Média = $z \n";
27
28 echo $a . $b; //concatenação
29
30 ?>
```

Pressione F11 para executar o código e teremos o seguinte resultado no console do Eclipse:



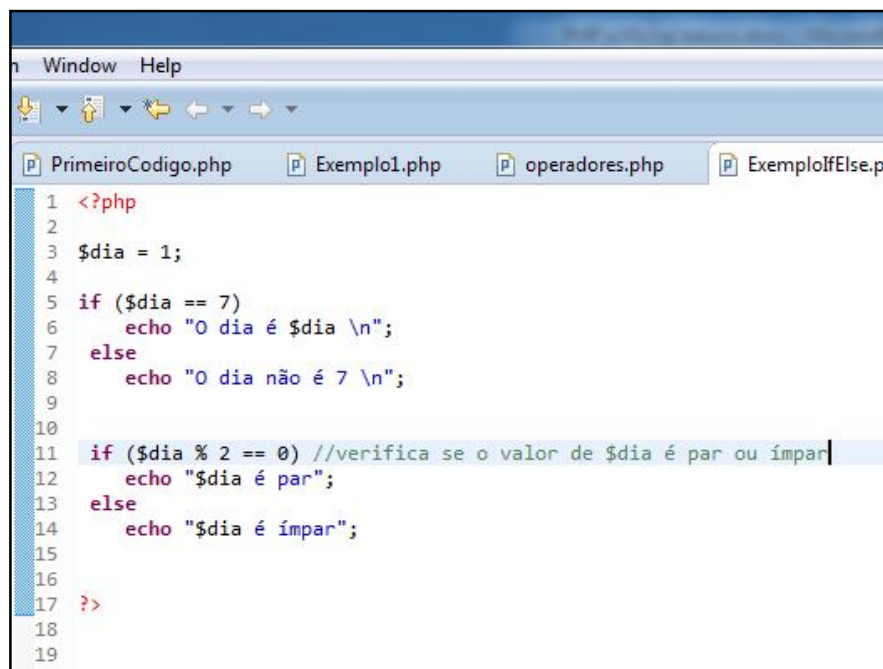


```
<terminated> operadores [PHP CLI Application] C:\xampp\php\php.exe
Soma = 12
Subtração = 2
Multiplicação = 35
Divisão = 1.4
Módulo = 2
Média = 8
Programação PHP
```

Perceba na linha 28 o operador de concatenação (.). Podemos entender concatenar, como juntar. Ele juntou os conteúdos das duas variáveis \$a e \$b.

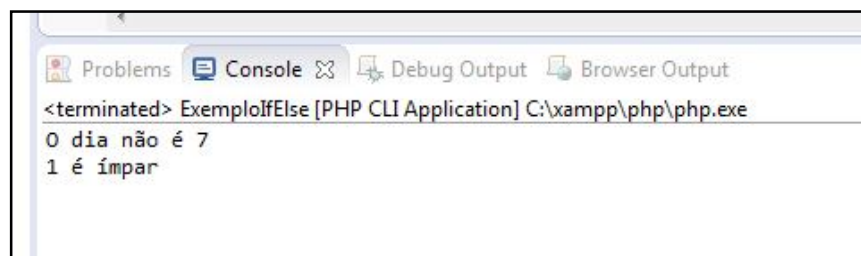
## :: Controle de Fluxo

- Instrução **If else** (se então): podemos ler como “se, condição verdadeira faça, senão faça outra coisa”, ou seja, o bloco de instrução só será executado se uma certa condição for satisfeita. Exemplo:



```
1 <?php
2
3 $dia = 1;
4
5 if ($dia == 7)
6     echo "O dia é $dia \n";
7 else
8     echo "O dia não é 7 \n";
9
10
11 if ($dia % 2 == 0) //verifica se o valor de $dia é par ou ímpar
12     echo "$dia é par";
13 else
14     echo "$dia é ímpar";
15
16
17 ?>
18
19
```

Pressione F11 para executar o código e teremos o seguinte resultado no console do Eclipse:

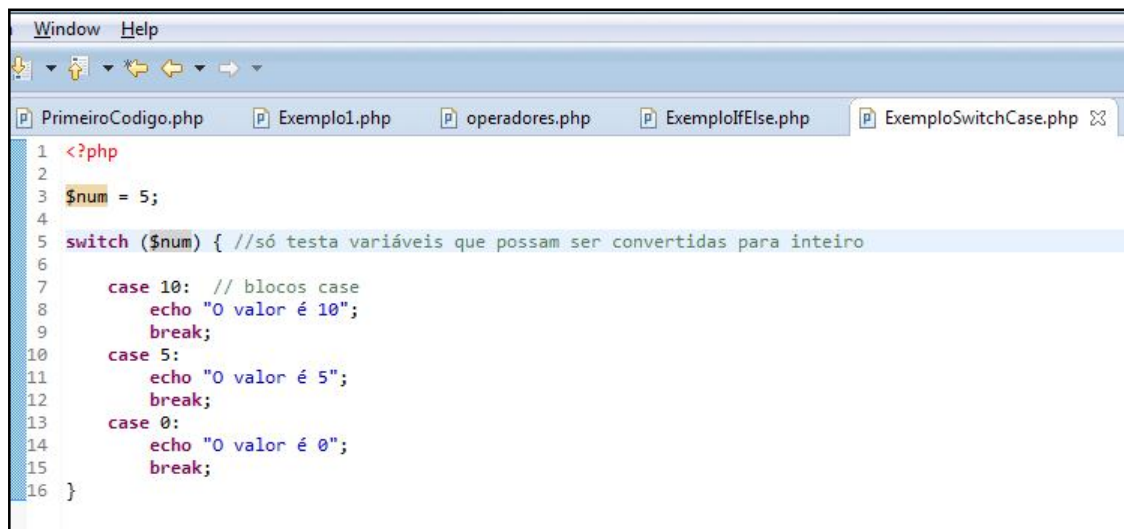


```
<terminated> ExemploIfElse [PHP CLI Application] C:\xampp\php\php.exe
O dia não é 7
1 é ímpar
```

Faça um teste também alterando o valor da variável dia para, por exemplo, 7, execute e veja o que acontece.

Faça o teste com outros valores para *\$dia* e você pode também salvar este arquivo com outro nome por exemplo "ControleFluxo.php". Menu File, Save As, digite o nome, Ok.

- Instrução **Switch** : esta instrução testa somente variáveis que possam ser convertidas para inteiro. Depois de testada a variável devemos trabalhar com blocos "**case**" dentro do switch. Exemplo:

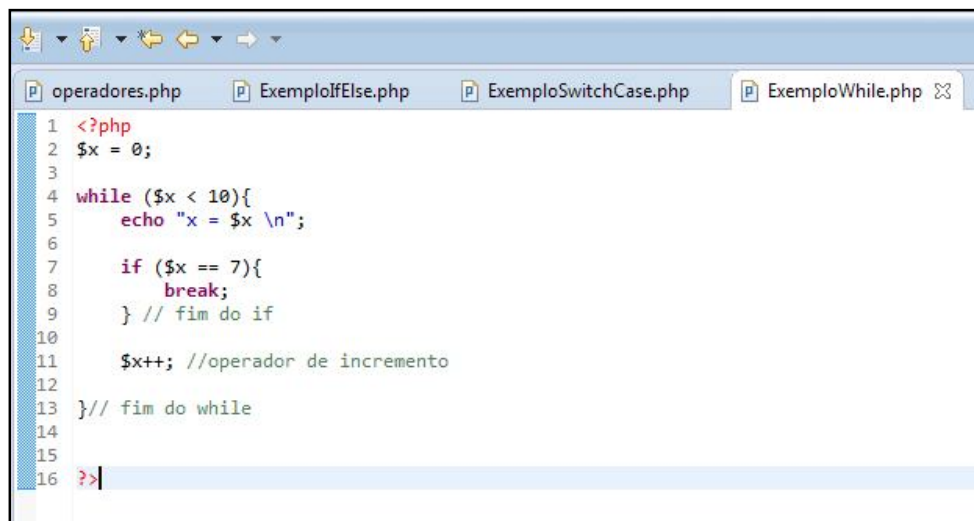


```
1 <?php
2
3 $num = 5;
4
5 switch ($num) { //só testa variáveis que possam ser convertidas para inteiro
6
7     case 10: // blocos case
8         echo "O valor é 10";
9         break;
10    case 5:
11        echo "O valor é 5";
12        break;
13    case 0:
14        echo "O valor é 0";
15        break;
16 }
```

Neste código o resultado será "o valor é 5".

Perceba a utilização em cada case do comando "**break**", ele serve para executar o case referente ao resultado da variável testada e logo depois sair do Switch, se você *não utilizar o break*, todos os cases serão executados. Teste este código no seu Eclipse atribuindo outros valores à variável *num*.

- Instrução **While** (enquanto): o bloco será executado *enquanto* a condição for verdadeira. Perceba a diferença entre o If else e o While. No primeiro o bloco será executado somente SE a condição for verdadeira; no segundo o bloco será executado ENQUANTO a condição for verdadeira. Exemplo:

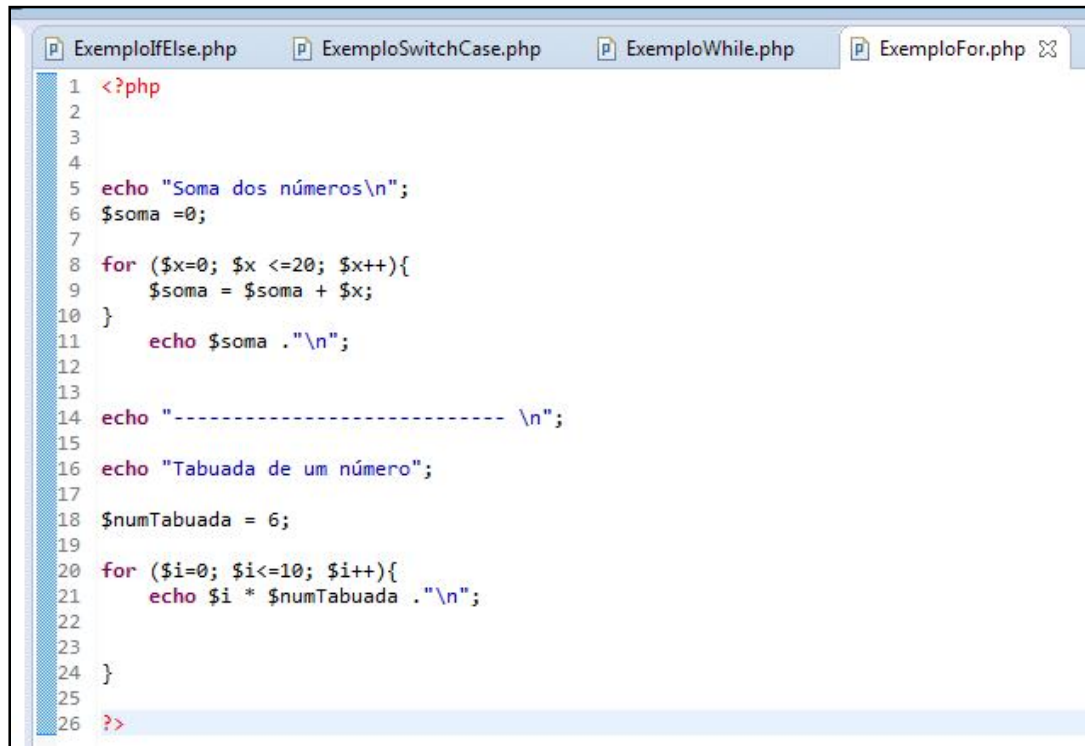


```
1 <?php
2 $x = 0;
3
4 while ($x < 10){
5     echo "x = $x \n";
6
7     if ($x == 7){
8         break;
9     } // fim do if
10
11     $x++; //operador de incremento
12
13 }// fim do while
14
15
16 ?>
```

Neste exemplo assim que o PHP entra no bloco while ele exibe o valor atual do x (que no caso é 0), testa se x é igual a 7, se x for igual a 7 interrompemos o bloco utilizando o **break** e o código é finalizado, se ainda não for igual a 7 ele é incrementado em uma unidade (com o operador de incremento **++**) e volta para o início do laço while. Se executar este código o resultado final de x será 7. Teste este código no Eclipse. Você pode também alterar a comparação no If de 7 para 9 por exemplo, só para testar.

- Instrução **For** (para): o bloco será executado até que atinja o valor recebido pela variável.

Exemplo:



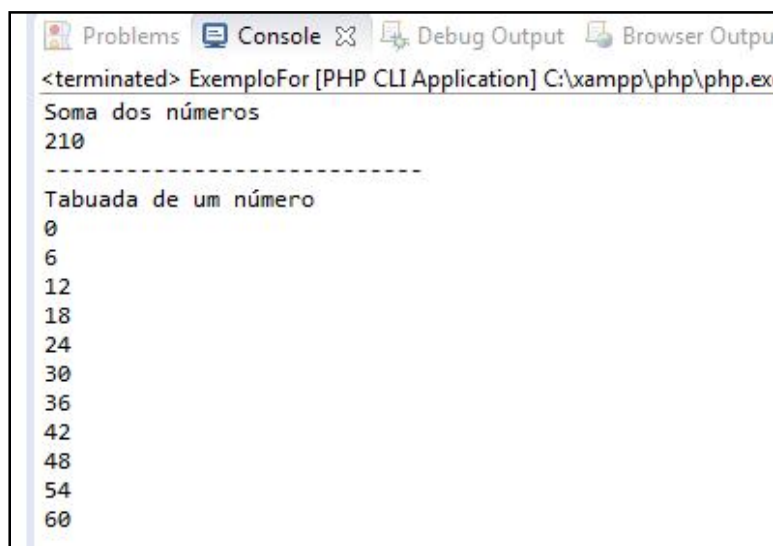
```
1 <?php
2
3
4
5 echo "Soma dos números\n";
6 $soma =0;
7
8 for ($x=0; $x <=20; $x++){
9     $soma = $soma + $x;
10 }
11     echo $soma ."\n";
12
13
14 echo "----- \n";
15
16 echo "Tabuada de um número";
17
18 $numTabuada = 6;
19
20 for ($i=0; $i<=10; $i++){
21     echo $i * $numTabuada ."\n";
22
23
24 }
25
26 ?>
```

Perceba que dentro dos parênteses do **For** (linha 8) temos 3 passos importantes separados por ponto e vírgula, são eles respectivamente: declaração da variável a ser testada com atribuição de um valor inicial ( \$x=0); o teste efetivo desta variável ( \$x <= 20); o seu incremento ( \$x++).

Dentro do primeiro código For, linha 9, a variável \$soma (que começou com o valor zero), vai receber o valor de \$soma + o valor de \$x atual. No final, linha 11, vai ser exibido o valor total da soma de 20 números.

No segundo código For, linha 20-21, ele vai exibir a tabuada de um número que é atribuído como valor inicial para a variável \$numTabuada.

Salve e execute este código (tecle F11) e terá o seguinte resultado:

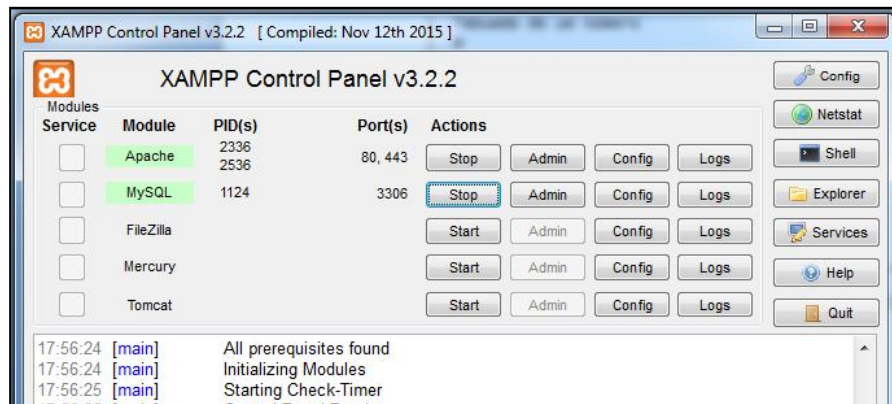


```
<terminated> ExemploFor [PHP CLI Application] C:\xampp\php\php.exe
Soma dos números
210
-----
Tabuada de um número
0
6
12
18
24
30
36
42
48
54
60
```

Faça testes. Altere o valor no primeiro For de 20 para 40 por exemplo. Ele vai exibir a soma dos 40 números. Altere o valor de \$numTabuada no segundo For.

### :: Módulo III - Criando e acessando um Banco de Dados MySQL com PHP

Quando você executa o xampp o Apache deve estar inicializado e o MySQL também, conforme figura abaixo.



Já vimos isso lá no começo da apostila na instalação do xampp.

O Apache é o nosso servidor web e o MySQL é o nosso banco de dados. É onde iremos armazenar informações, ele vai garantir a nossa persistência de dados, ou seja, em um sistema de cadastro por exemplo, é onde essas informações ficaram gravadas, em uma tabela MySQL.

Assim como o Apache, o MySQL também já vem com o Xampp, então basta ele estar inicializado para podermos começar.

Vamos criar agora o nosso primeiro Banco de Dados.

1 – Entre no seu navegador e acesse o endereço localhost (127.0.0.1) e tecla Enter, você verá a tela de boas vindas do Xampp.

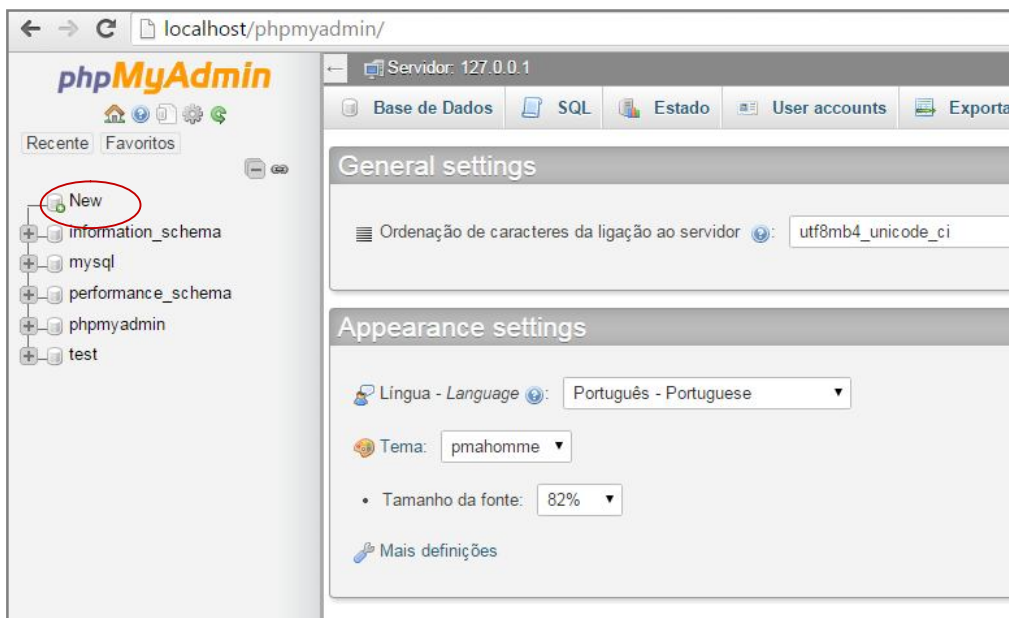


2 – Agora clique na opção **phpMyAdmin**. É aqui que vamos criar nosso banco de dados.

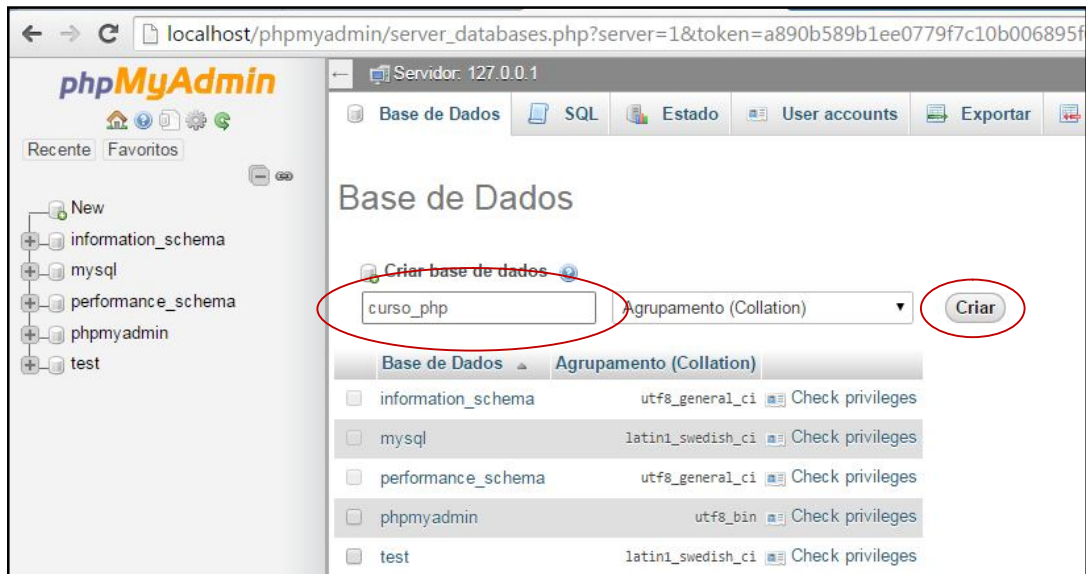
Obs: depois vamos aprender comandos SQL onde poderemos fazer operações direto pelo Eclipse.



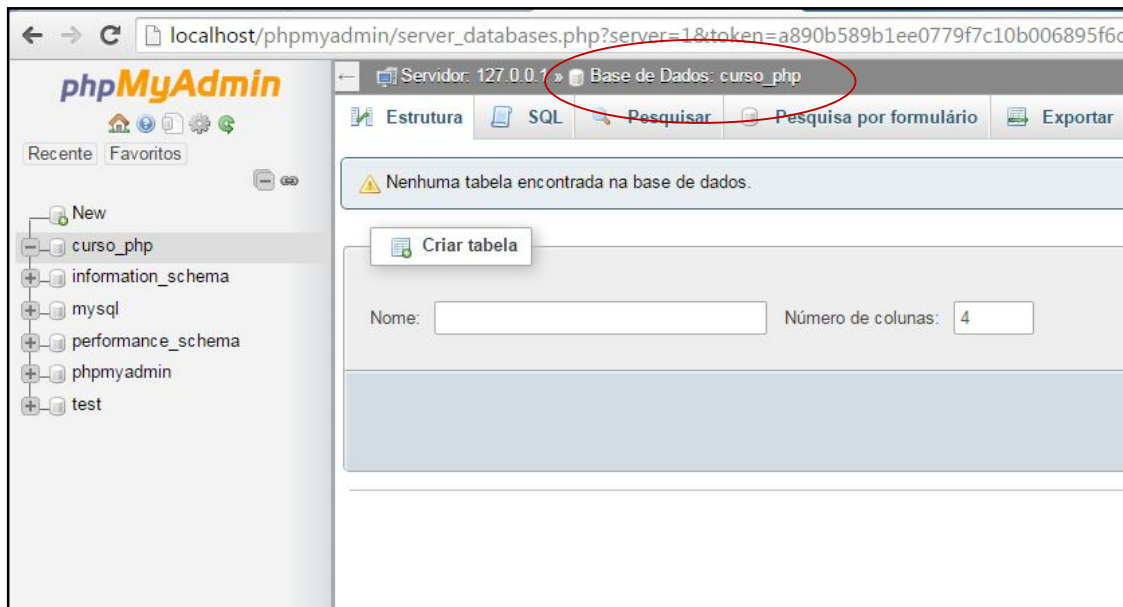
3 – Após clicar na opção phpMyAdmin você terá a tela da figura abaixo. Clique em New.



4 – Na opção *Criar base de dados* digite o nome do nosso banco de dados, **curso\_php** e clique em *Criar*.



5 – Pronto nosso banco de dados **curso\_php** está criado.



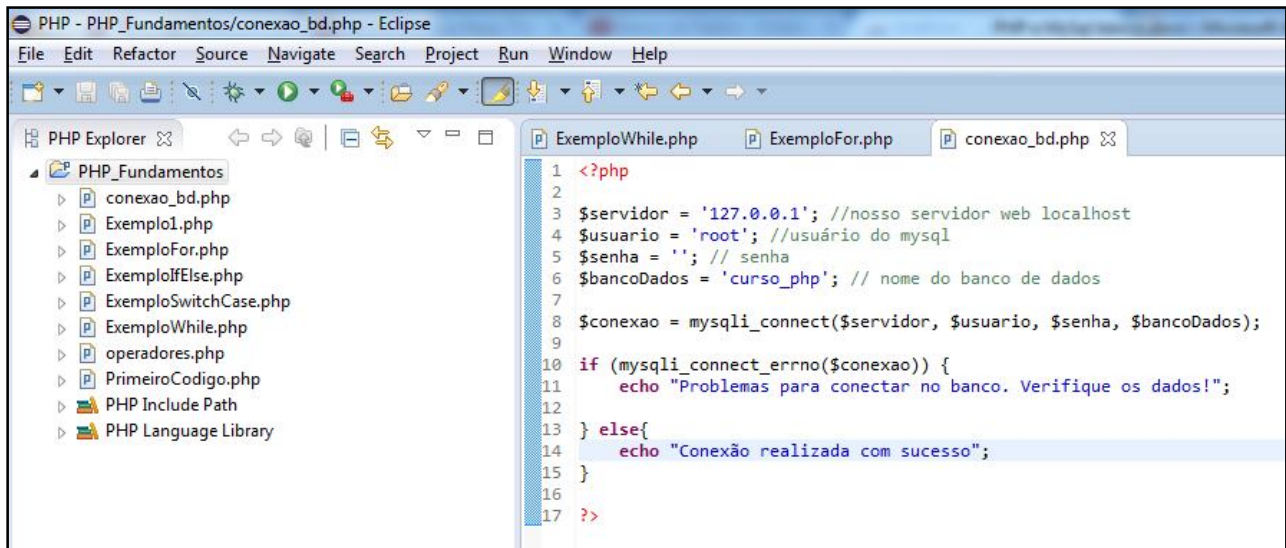
O **phpMyAdmin** que é um aplicativo web muito famoso e muito usado para administrar o MySQL via navegador.

## :: Conexão com o banco de dados

Agora vamos fazer a conexão do banco de dados via linha de comando em um código PHP usando o Eclipse. Vamos utilizar o usuário e senha do próprio MySQL. Neste momento estamos misturando tudo: programando em PHP na IDE Eclipse e acessando o banco de dados MySQL criado no phpMyAdmin. Legal né rsrsrs

No Eclipse crie um arquivo chamado **conexao\_bd.php**. Este arquivo pode ter qualquer nome, você escolhe. Digite o código abaixo para testarmos a conexão com o nosso banco de dados.





Quando você executar este código deverá ver a mensagem “Conexão realizada com sucesso”. Vamos entender o que está ocorrendo neste código.

- Linha 3: uma variável \$servidor que recebe o endereço do servidor utilizado. Nesse caso 127.0.0.1 que também poderia ser substituído por localhost sem problemas. Digamos que você tenha um domínio na Internet e esteja fazendo o seu próprio site, este servidor seria algo parecido com [HTTP://www.seudominio.com](http://www.seudominio.com).
- Linha 4: o usuário que terá acesso ao banco de dados, nesse caso utilizei o usuário root do próprio MySQL.
- Linha 5: a senha deste usuário é vazia, campo sem senha.
- Linha 6: variável \$bancoDados que recebe o nome do banco de dados a ser acessado.

Obs: os nomes das variáveis podem ser qualquer nome, o que importa mesmo é o conteúdo dessas variáveis.

- Linha 8: variável \$conexao que recebe a função de conexão de banco de dados mysqli\_connect. Esta função recebe como parâmetros as variáveis com os dados de acesso ao banco. Detalhe: a ordem dos parâmetros é obrigatoriamente “servidor, usuário, senha, nome do banco”, não se pode alterar esta ordem.
- Linha 10: um IF com a função mysqli\_connect\_errno. Esta função testa se a conexão foi bem sucedida. Essa função recebe como parâmetro justamente a variável \$conexao que criamos para receber as configurações do banco, se ocorrer algum erro no meio da execução irá aparecer a mensagem da linha 11, se ocorrer tudo bem então aparece a mensagem da linha 14.

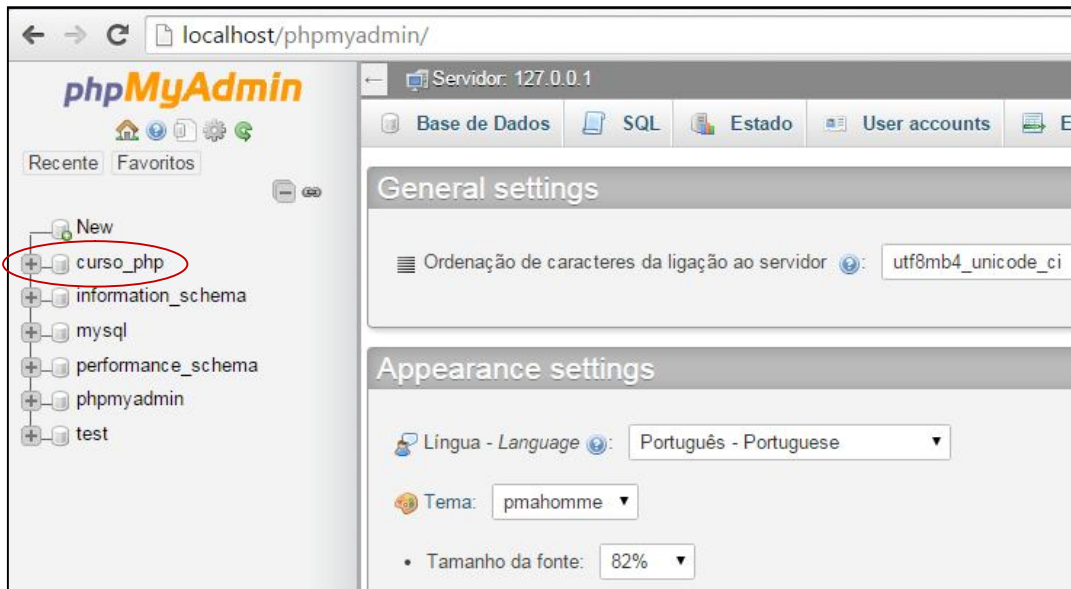
Para testar, experimente trocar o usuário e executar o código, você verá que a conexão não será realizada.

## :: Módulo IV - Sistema Usuário e Senha

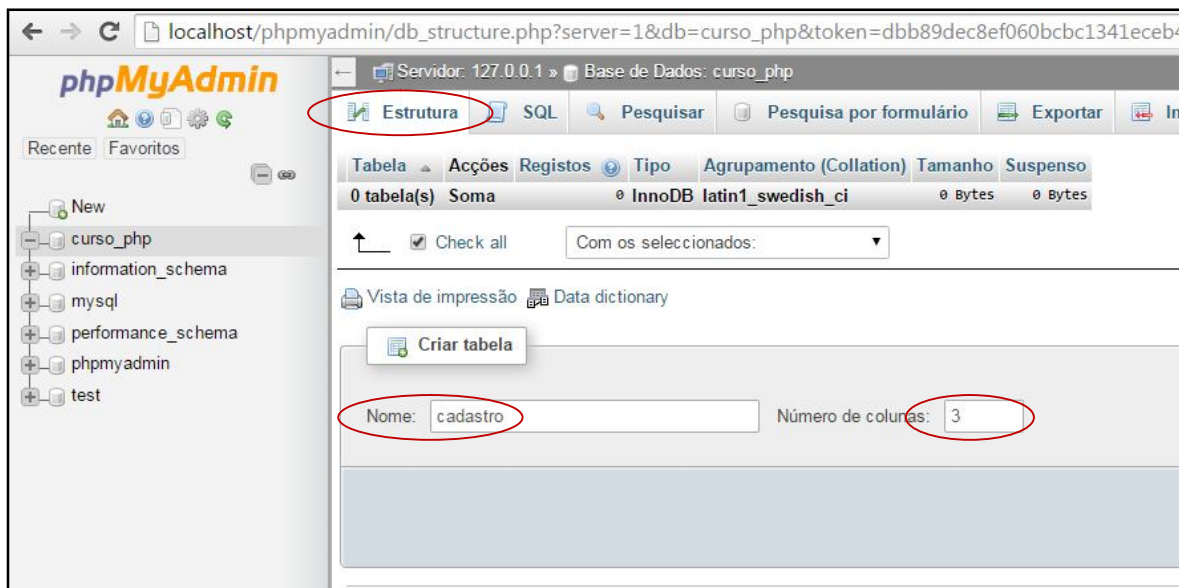
Agora vamos que já criamos e conectamos nosso banco de dados, vamos criar um pequeno sistema para cadastrar usuário e senha em uma tabela mysql.

Obs: vamos precisar utilizar um pouco de HTML também. Tudo será explicado aqui na apostila, mas se você quiser pesquisar um básico de HTML é bom também.

1 – Vamos acessar o phpMyAdmin para criar nossa tabela mysql. Acesse no seu navegador o endereço <http://localhost/phpmyadmin/> e tecla enter. (lembrando que já criamos o banco de dados). Clique no nosso banco de dados curso\_php.



2 - Em Estrutura digite o nome da nossa tabela "cadastro" e em Número de colunas digite 3, que é a quantidade de campos da tabela (id, login e senha). Clique em Executar.



3 – Ao Executar você terá uma tabela chamada "cadastro" que conterá 3 campos. Temos que configurar esses campos da seguinte maneira:

- Campo id: Este campo será do tipo int (inteiro e será também a nossa chave primária. Resumidamente chave primária é um campo único de uma tabela, onde este campo terá um valor único que não pode ser repetido). Também deve deixar marcada a opção de autoincremento.



admin/db\_structure.php?server=1&db=curso\_php&token=dbb89dec8ef060bcbc1341eceb4877a5

Servidor: 127.0.0.1 » Base de Dados: curso\_php » Tabela: cadastro

Table name: cadastro Add 1 column(s) Executar

Nome	Tipo	Tamanho/Valores*	Predefinido	Agrupamento (Collation)	Atributos	Nulo	Índice	A_I
id	INT		None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
login	VARCHAR	20	None			<input type="checkbox"/>	---	<input type="checkbox"/>
senha	VARCHAR	8	None			<input type="checkbox"/>	---	<input type="checkbox"/>

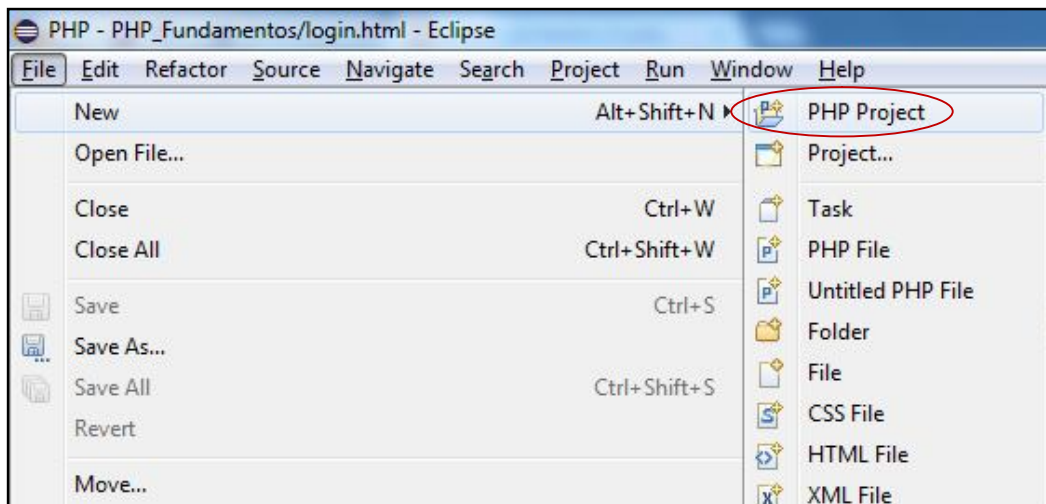
- Campo login: Este campo irá receber o nome do usuário, vai ser do tipo varchar (é como o String no PHP) e terá tamanho 20 (pode ter no máximo 20 caracteres).

- Campo senha: Este campo irá receber a senha do usuário, vai ser do tipo varchar e terá tamanho 8. Clique em Guarda. Pronto, temos nossa tabela criada.

Pré-visualização SQL Guarda

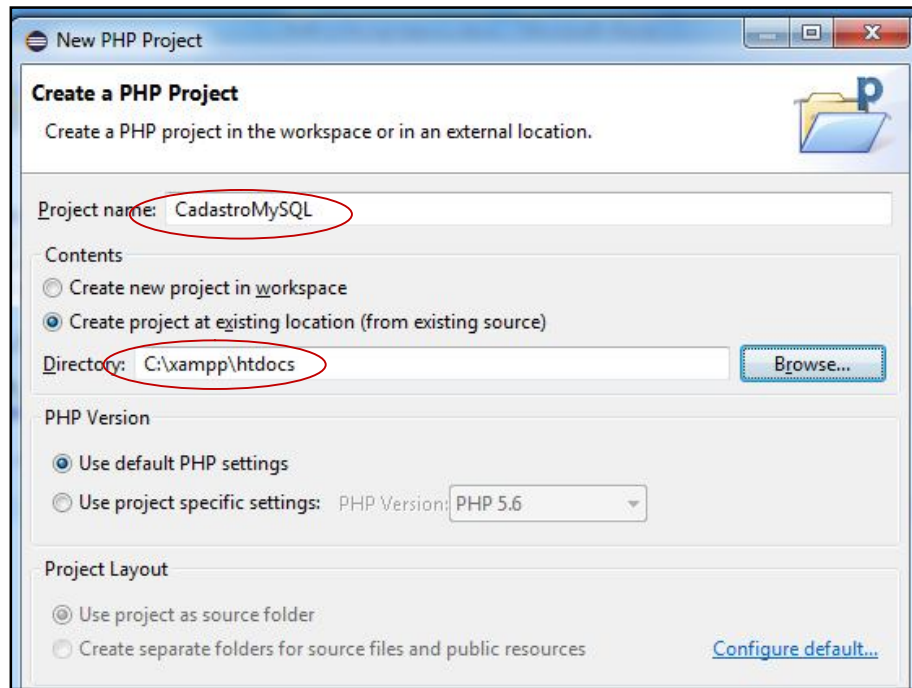
Dica: aqui nós estamos utilizando o phpMyAdmin para criar nossa tabela, mas você pode ter acesso ao código SQL utilizado clicando em Pré-visualização SQL, teste que é bem interessante ter acesso ao código "cru" da tabela.

4 – Voltando ao Eclipse, crie um novo projeto para ser salvo na pasta específica do xampp. File, New, PHP Project

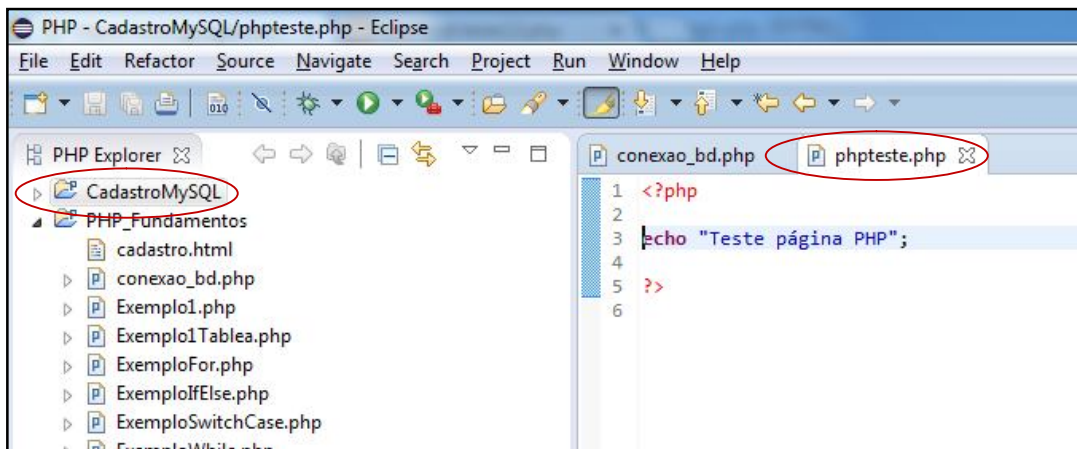


5 – Agora preencha o campo Project name com um nome para o seu projeto (eu utilizei CadastroMySQL) e Directory com o diretório do xampp conforme figura abaixo. Clique em Finish.

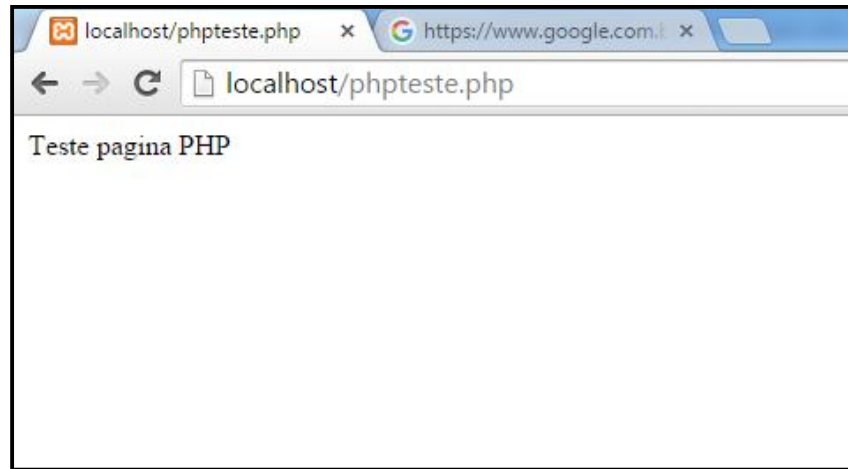
Obs: essa configuração é para podermos fazer testes via navegador e é muito importante estar na pasta htdocs do xampp, é onde o navegador vai procurar o arquivo a ser executado .



6 – Vamos testar para saber se está tudo ok. Crie um arquivo chamado phpteste.php dentro desse novo projeto que acabamos de criar. Botão direito no projeto CadastroMySQL, New, PHP File e coloque o nome do arquivo phpteste.php. Digite uma mensagem qualquer como na figura.



7 – Abra o seu navegador e digite o endereço <http://localhost/phpteste.php> e você estará acessando sua página conforme a figura.

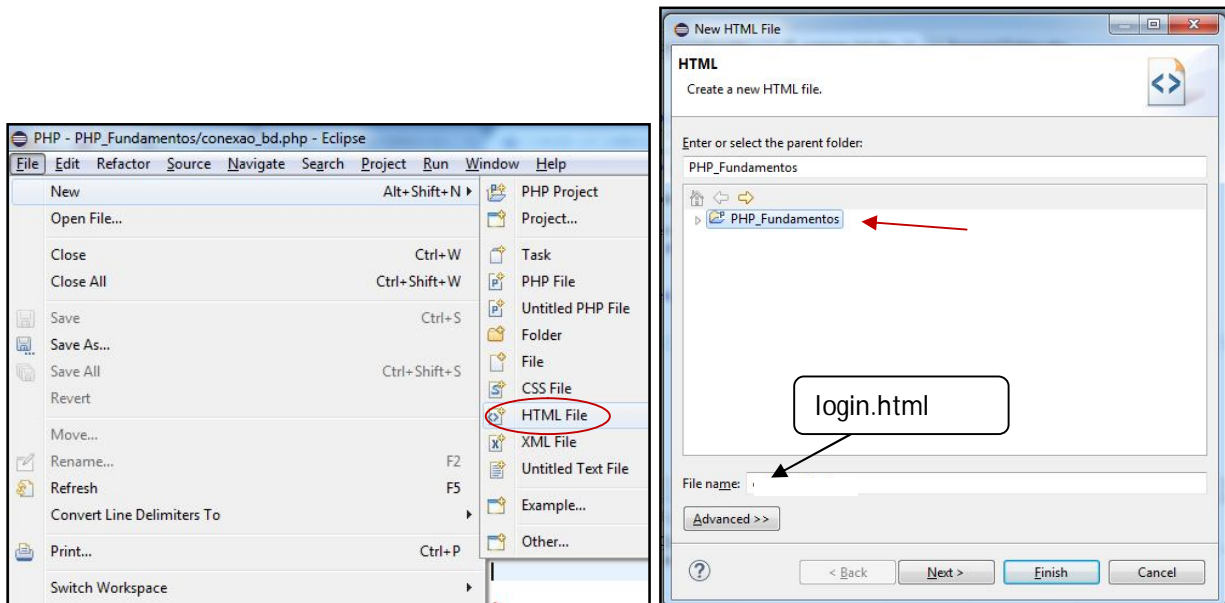


Ótimo nosso PHP ta rodando normalmente no navegador (armazenando os arquivos php automaticamente no htdoc do xampp) e agora sim vamos criar nosso Cadastro de usuário com PHP/MySQL.

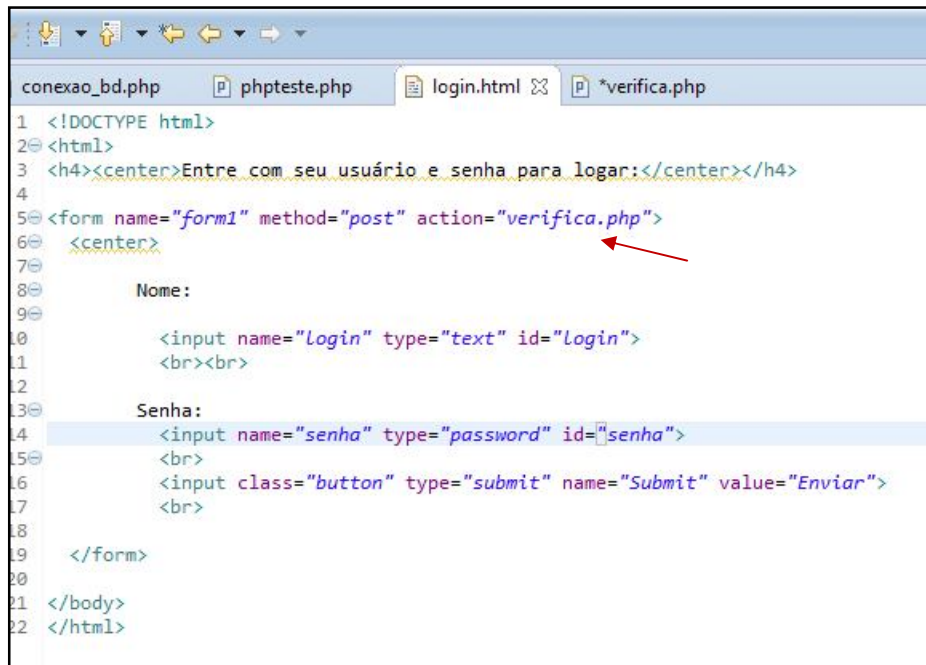
Obs: Como o nosso curso é de PHP, então eu não vou me aprofundar no HTML, mas adianto a você que é importante ter ao menos uma base de HTML (é uma linguagem fácil) para qualquer tipo de programação para web. No nosso caso estamos aprendendo PHP/MySQL, mas o HTML por vezes se torna indispensável.

8 – Voltando ao Eclipse, vamos criar agora o nosso Formulário de login, este arquivo será um HTML e não um PHP. Calma que daqui a pouco vai clarear e você entenderá tudo rsrsr.

- No Eclipse, clique em File, New, HTML File, clique em nossa pasta CadastroMySQL e digite o nome do arquivo login.html.



Digite o código abaixo.

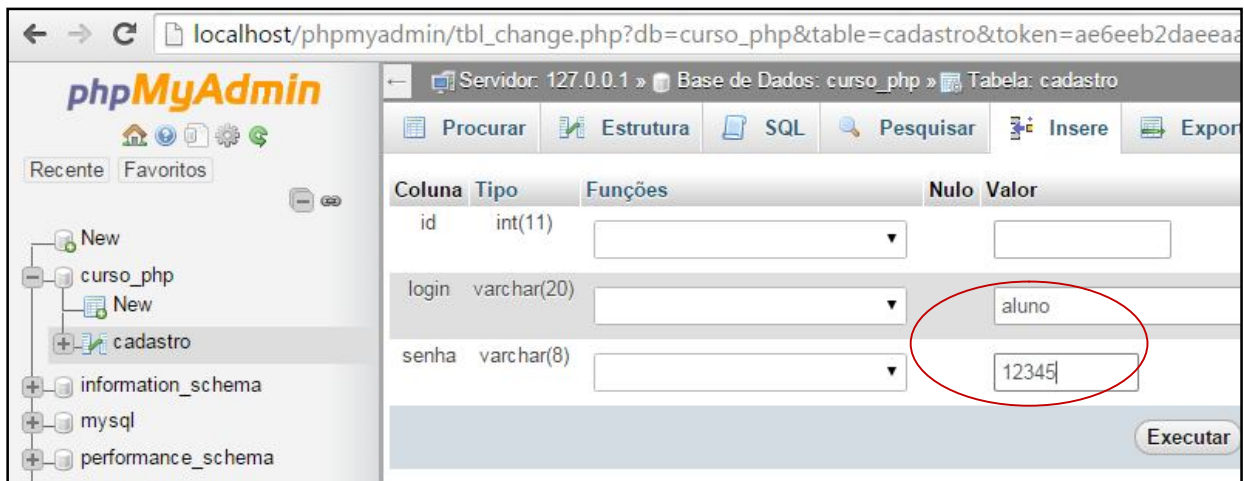


```
1 <!DOCTYPE html>
2 <html>
3 <h4><center>Entre com seu usuário e senha para logar:</center></h4>
4
5 <form name="form1" method="post" action="verifica.php">
6   <center>
7
8     Nome:
9
10    <input name="login" type="text" id="login">
11    <br><br>
12
13    Senha:
14    <input name="senha" type="password" id="senha">
15    <br>
16    <input class="button" type="submit" name="Submit" value="Enviar">
17    <br>
18  </form>
19
20 </body>
21 </html>
```

Aqui temos um código HTML com um form que irá enviar as informações do campo login e senha para o arquivo verifica.php que irá verificar se os dados do usuário estão corretos.

9 – Antes de criarmos o arquivo verifica.php, precisamos cadastrar um usuário e senha no phpMyAdmin para o sistema poder realizar a consulta na tabela cadastro.

Acesse o phpMyAdmin em <http://localhost/phpmyadmin>, na coluna da esquerda clique no nosso banco de dados curso\_php, depois clique na tabela cadastro, na parte superior clique na opção **Inserir** e digite o nome e senha conforme figura abaixo e clique em Executar, pronto nosso usuário “aluno” com a senha “12345” está cadastrado e podemos voltar ao Eclipse.



10 – Vamos criar o arquivo verifica.php, ele irá consultar nosso banco de dados MySQL (curso\_php) para verificar se o login e senha são válidos. No passo 8 criamos um arquivo HTML, agora vamos criar um arquivo PHP normalmente como já fizemos anteriormente aqui no curso. Crie um arquivo chamado verifica.php e digite o código a seguir. Lembrando que o arquivo também deve estar no projeto CadastroMySQL.



```
1 <?php
2 //aproveitando o código conexao_bd que já fizemos
3
4 $servidor = '127.0.0.1'; // nosso servidor web localhost
5 $usuario = 'root'; // usuário do mysql
6 $senha = ''; // senha
7 $bancoDados = 'curso_php'; // nome do banco de dados
8
9 $conexao = mysql_connect ( $servidor, $usuario, $senha, $bancoDados ) or die(mysql_error());
10
11 mysql_select_db('curso_php', $conexao); //seleciona o banco curso_php
12
13 if (mysqli_connect_errno ( $conexao )) {
14     echo "Problemas para conectar no banco. Verifique os dados!";
15 } else {
16     echo "Conexao realizada com sucesso";
17 }
18
19 $login = $_POST["login"];
20 $senha = $_POST["senha"];
21
22
23 $selecao = mysql_query("SELECT * FROM cadastro WHERE login = '$login' AND senha = '$senha' ");
24
25 $row = mysql_fetch_array($selecao);
26
27 if ($row == "")
28 {
29     echo "<br>login e/ou senha inválidos.</center>";
30     echo "<br><br>";
31     echo "Volte e tente novamente";
32     exit;
33 } else {
34     echo "<br><br>Bem vindo(a) <b>$login</b>";
35 }
36
37 ?>
```

Explicando o código:

- Linhas 4 a 7: variáveis com as informações de conexão do banco de dados, já testamos isso no código conexao\_bd.php;
- Linha 9: variável \$conexao que recebe a função **mysql\_connect**, esta função é que irá realizar a conexão com o MySQL recebendo como parâmetros as variáveis de conexão definidas anteriormente, se a conexão falhar, então entra em ação a função **mysql\_error** exibindo um erro de conexão;
- Linha 11: com o MySQL devidamente conectado, agora temos que selecionar o banco de dados que iremos utilizar, nesse caso o curso\_php. Para isso utilizamos a função **mysql\_select\_db** passando como parâmetro o nome do banco de dados (curso\_php) e a conexão realizada (\$conexao);
- Linha 13 a 17: um IF que testa a conexão do banco utilizando a função **mysqli\_connect\_errno** que também já vimos no conexao\_bd.php;
- Linha 23: variável \$selecao que recebe a função **mysql\_query**, esta função irá fazer a consulta mysql em nosso banco de dados. Dentro dos parênteses nós temos basicamente código SQL e não PHP, estamos falando para o MySQL mais ou menos isso "selecione todos os registros da tabela cadastro onde login = \$login e senha = \$senha.

Obs: o campo login e senha vieram lá do arquivo login.html e foram preenchidos pelo usuário. Estes campos são atribuídos às variáveis \$login e \$senha do arquivo verifica.php. Já dentro do comando SQL da linha 23, login e senha (em azul) são campos da tabela cadastro lá do MySQL que estão sendo comparados com as

variáveis \$login e \$senha do arquivo verifica.php.

```
23 $selecao = mysql_query("SELECT * FROM cadastro WHERE login = '$login' AND senha = '$senha' ");
```

- Linha 25: variável \$row que recebe a função **mysql\_fetch\_array** que retorna a linha consultada na tabela, no caso login e senha da tabela cadastro. Esta função recebe como parâmetro a consulta do comando select.

- Linha 27: um IF que vai testar se o usuário e senha estão corretos. Se \$row for vazio, é sinal que a consulta não gerou nenhum resultado e nesse caso o IF retorna uma mensagem de usuário e senha inválidos.

Senão, ou seja, se \$row gerou algum resultado, significa que o usuário e senha são válidos, eles foram encontrados na tabela cadastro do banco de dados curso\_php, o IF gera uma mensagem de bem vindo e acessa o sistema normalmente.

Obs: perceba alguns comandos HTML como <br> sendo usados entre aspas, isso é perfeitamente usual. <br> é linha em branco em HTML.