**PHP - Boas práticas e conceitos modernos**

**João Ribeiro - Sys4soft**

<https://www.youtube.com/watch?v=krQj4stNu5U&list=PLXik_5Br-zO_5EGPG6_u-u0hVI_f_ThO_>

Resumo do curso feito por Roberto Pinheiro

# Aula 01 - Instalação do PHP no PC

- Acesse a URL https://www.php.net/downloads.php

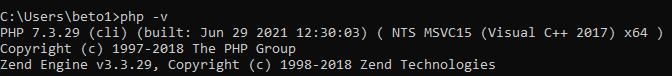
- Baixe a versão 7.3.29 do PHP para Windows em formato zip.

- Crie a pasta C:\php e extraia o conteúdo do arquivo zip para dentro dela.

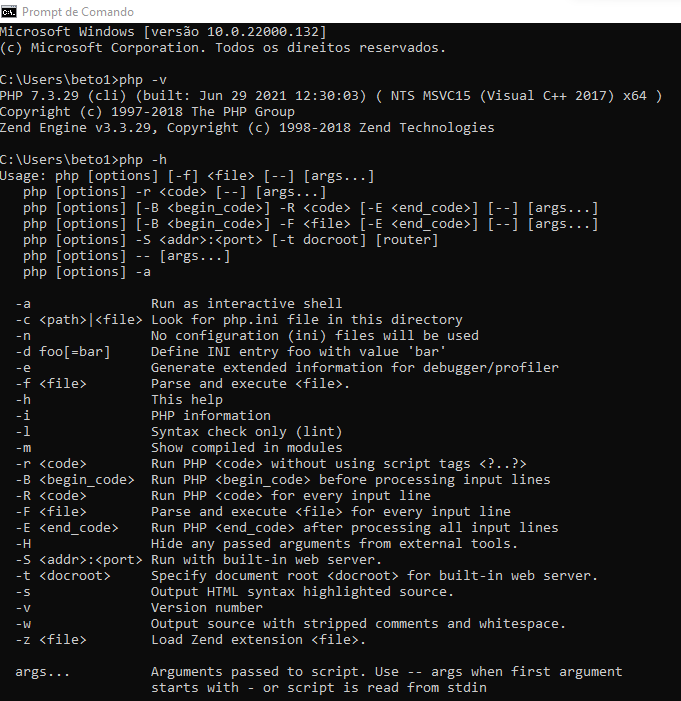
- Crie a pasta C:\www. Esta será a pasta do projeto, ou seja, onde iremos armazenar os scripts em PHP.

- Em Definições Avançadas do Sistema, clique em variáveis de ambiente, e defina o path onde foi instalado o PHP.

php -v

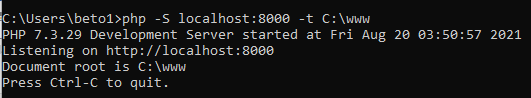


php -h



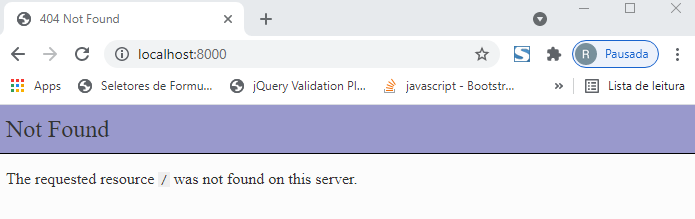
## Iniciando o servidor PHP

php -S localhost:8000 -t C:\www



- No browser:

localhost:8000



- Na pasta do projeto crie um arquivo chamado index.php:

**C:\www\index.php**

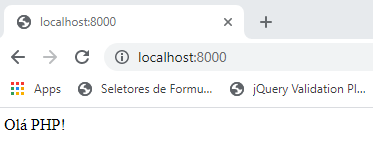
<?php

echo 'Olá PHP!';

?>

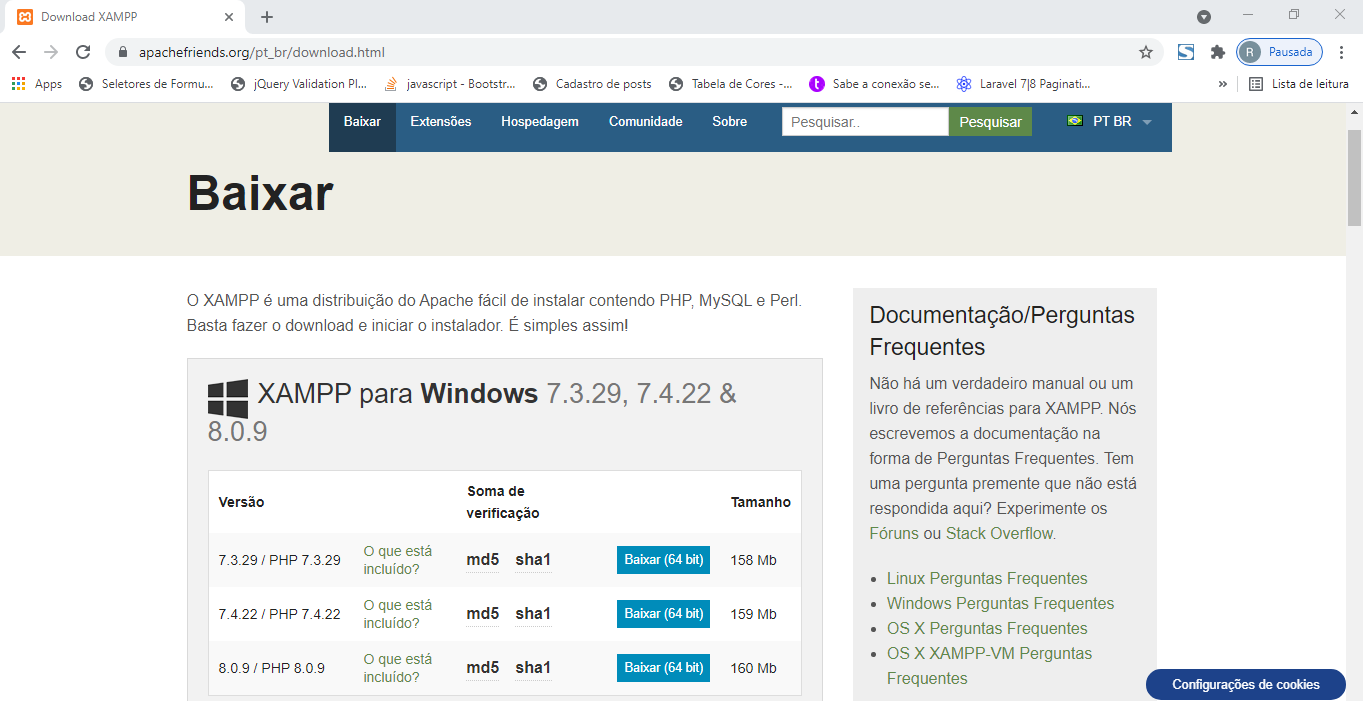
- No browser:

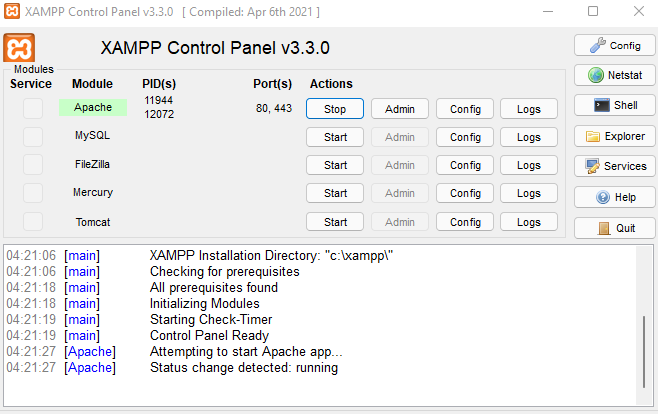
localhost:8000



# Aula 02 - Utilização do PHP com o XAMPP

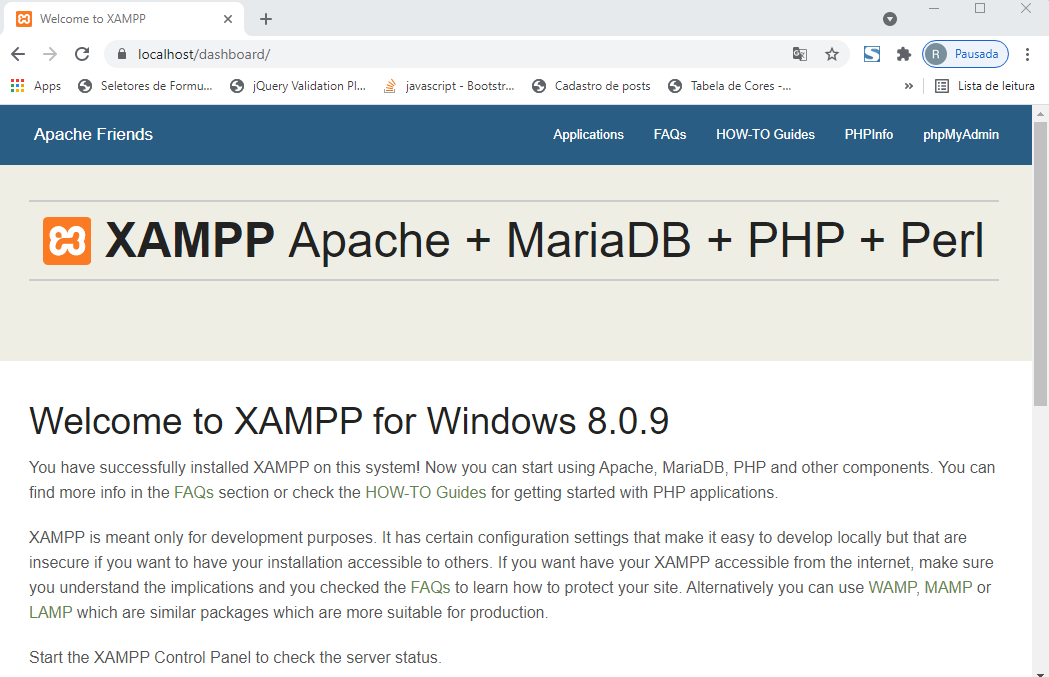
- Baixe e instale o XAMPP para Windows.





- No browser:

localhost



- Dentro da pasta C:\xampp\htdocs crie a pasta php\_sys4soft e dentro dela o arquivo index.php

**C:\xampp\htdocs\aula\_02\index.php**

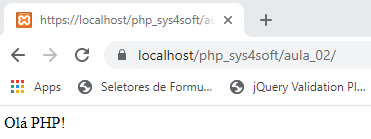
<?php

echo 'Olá PHP!';

?>

- No browser:

https://localhost/php\_sys4soft/aula\_02/

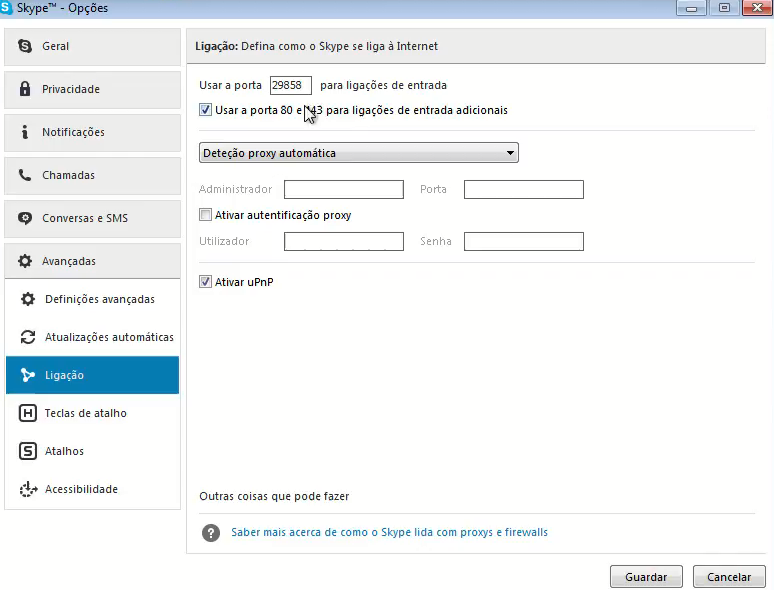


# Aula 04 - Resolvendo conflito entre skype e xampp

- Na forma padrão, ao se utilizar o Skype, é gerado um conflito de porta com o Xampp, o que não permite utilizá-lo.

- No Skype, entre em Ferramentas e em Opções:

- Selecione Avançadas:



- Desmarque a caixa Usar a porta 80 e 443 para ligações de entrada adicionais.

- Salve as configurações.

# Aula 05 - Alterando as portas nas configurações do XAMPP

## Alterando de porta 80 para porta 8080

- No XAMPP, clique no botão Config do Apache e selecione a opção Apache (httpd.conf):

- Procure por 80.

- Substitua Listen 80 por Listen 8080.

- Substitua Server Name localhost 80 por Server Name localhost 8080.

- Salve o documento

## Alterando de porta 443 para porta 44343

- No XAMPP, clique no botão Config do Apache e selecione a opção Apache (httpd.ssl.conf):

- Procure por 443.

- Substitua Listen 443 por Listen 44343.

- Substitua <VirtualHost \_default\_:443> por <VirtualHost \_default\_:44343>.

- Salve o documento

# Aula 06 - utilização correta das tags PHP para abrir e fechar blocos de codigo

**C:\xampp\htdocs\aula\_06\index.html**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF8">

<title>Título da página</title>

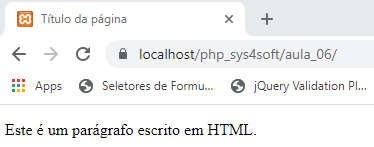
</head>

<body>

<p>Este é um parágrafo escrito em HTML.</p>

</body>

</html>



## Código PHP dentro de uma estrutura HTML

**C:\xampp\htdocs\aula\_06\index.php**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF8" />

<title>Título da página</title>

</head>

<body>

<p>Este é um parágrafo escrito em HTML.</p>

<?php

echo 'Texto processado em PHP.';

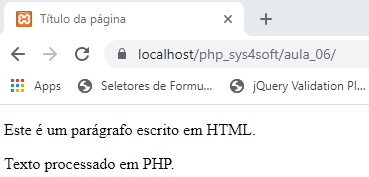
?>

</body>

</html>

- No browser:

https://localhost/php\_sys4soft/aula\_06/



## Código PHP puro

**C:\xampp\htdocs\aula\_06\index1.php**

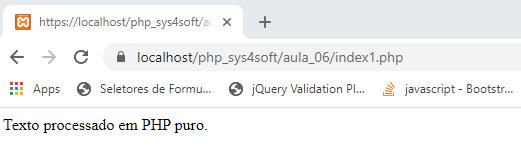
<?php

echo 'Texto processado em PHP puro.';

?>

- No browser:

https://localhost/php\_sys4soft/aula\_06/index1.php



## Tags para se escrever em PHP

<?php ?>

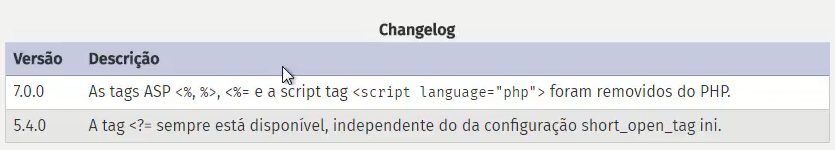
<?= ?>

<? ?> Válidas até a versão 5.4. Descontinuadas a partir da versão 7.0

<% %> Válidas até a versão 5.4. Descontinuadas a partir da versão 7.0

Apesar de não recomendado, as duas últimas opções podem se tornar válidas no PHP 7, alterando o arquivo php.ini. Para isso, a linha que contém short\_open\_tag=Off deve ser alterada para short\_open\_tag=On.

Vale lembrar que em um servidor de internet, onde o PHP é compartilhado entre os usuários (grande maioria dos casos) não é permitido alterar o arquivo php.ini.



# Aula 07 - Exemplo prático da utilização do include

- O include permite inserir código de um arquivo PHP à parte dentro de outro arquivo PHP.

**aula\_07\header.php**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF8" />

<title>Include</title>

</head>

<body>

<!-- cabeçalho -->

<div>

<a href="index.php">Início</a> |

<a href="servicos.php">Serviços</a> |

<a href="contatos.php">Contatos</a> |

<a href="links.php">Links</a> |

<hr />

</div>

**aula\_07\footer.php**

<!-- rodapé -->

<div>

<hr />

<p>Empresa @copy 2021</p>

<p><a href="mailto: empresa@gmail.com">empresa@gmail.com</a></p>

</div>

</body>

</html>

**aula\_07\servicos.php**

<?php include('header.php') ?>

<!-- Conteúdo -->

<div>

<h1>Serviços</h1>

</div>

<?php include('footer.php') ?>

**aula\_07\contatos.php**

<?php include('header.php') ?>

<!-- Conteúdo -->

<div>

<h1>Contatos</h1>

</div>

<?php include('footer.php') ?>

**aula\_07\links.php**

<?php include('header.php') ?>

<!-- Conteúdo -->

<div>

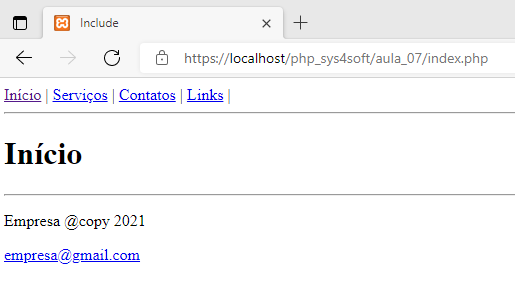
<h1>Links úteis</h1>

</div>

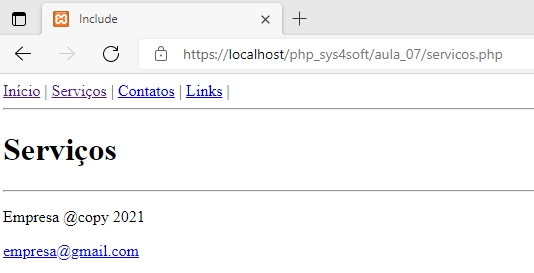
<?php include('footer.php') ?>

- No browser

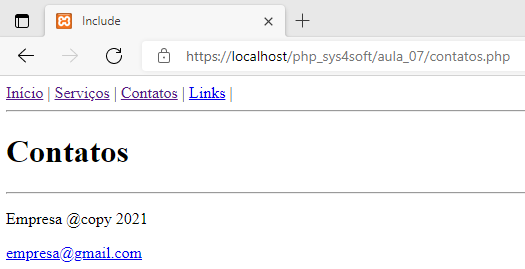
https://localhost/php\_sys4soft/aula\_07/index.php



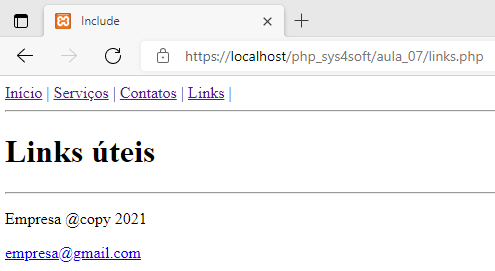
https://localhost/php\_sys4soft/aula\_07/servicos.php



https://localhost/php\_sys4soft/aula\_07/contatos.php



https://localhost/php\_sys4soft/aula\_07/links.php



# Aula 08 - Diferença entre include e include\_once

## Uso de include

**aula\_08\index.php**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF8" />

<title>include e include\_once</title>

</head>

<body>

<p>Este é um parágrafo escrito em HTML.</p>

<?php include('teste.php') ?>

</body>

</html>

**aula\_08\teste.php**

<?php

function teste(){

echo 'Este é um texto escrito em PHP';

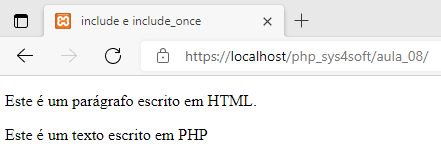
}

teste();

?>

- No browser:

https://localhost/php\_sys4soft/aula\_08/



**aula\_08\index.php**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF8" />

<title>include e include\_once</title>

</head>

<body>

<p>Este é um parágrafo escrito em HTML.</p>

<?php include('teste.php') ?>

<?php include('teste.php') ?>

<?php include('teste.php') ?>

<?php include('teste.php') ?>

</body>

</html>

**aula\_08\teste.php**

<?php

function teste(){

echo 'Este é um texto escrito em PHP';

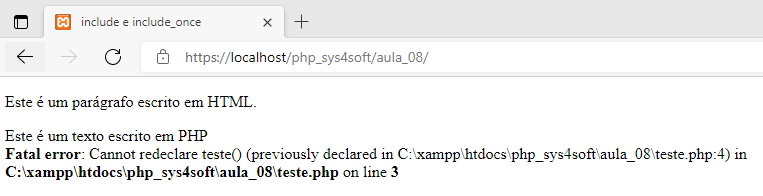
}

teste();

?>

- No browser:

https://localhost/php\_sys4soft/aula\_08/



## Uso de include\_once

**aula\_08\index.php**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF8" />

<title>include e include\_once</title>

</head>

<body>

<p>Este é um parágrafo escrito em HTML.</p>

<?php include\_once('teste.php') ?>

<?php include\_once('teste.php') ?>

<?php include\_once('teste.php') ?>

<?php include\_once('teste.php') ?>

</body>

</html>

- Irá executar apenas uma vez a função

**aula\_08\teste.php**

<?php

function teste(){

echo 'Este é um texto escrito em PHP';

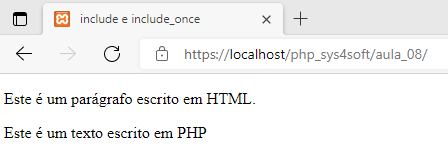
}

teste();

?>

- No browser:

https://localhost/php\_sys4soft/aula\_08/



# Aula 09 - Diferença entre uso de double quotes e single quotes

**aula\_09\index.php**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF8" />

<title>Aspas simples e aspas duplas</title>

</head>

<body>

<p>

<?php echo '<h3>Esta frase usa single quotes.</h3>' ?>

</p>

<p>

<?php echo "<h3>Esta frase usa double quotes.</h3>" ?>

</p>

<p>======================================</p>

<p>

<?php

$nome1 = 'João';

echo 'Nome: $nome1';

?>

</p>

<p>

<?php

$nome2 = 'Maria';

echo "Nome: $nome2";

?>

</p>

<p>======================================</p>

<p>

<?php

echo 'O\'Reilly';

?>

</p>

<p>

<?php

echo "O'Reilly";

?>

</p>

<p>======================================</p>

<p>

<?php

echo 'texto \n texto';

?>

</p>

<p>

<?php

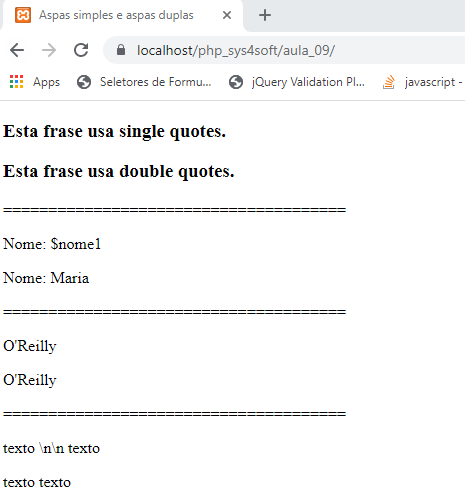
echo "texto \n texto";

?>

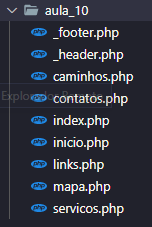
</p>

</body>

</html>



# Aula 10 - Exercício muito simples de direcionamento em PHP



**aula\_10\\_header.php**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF8" />

<title>PHP</title>

</head>

<body>

<!-- cabeçalho -->

<div>

<a href="index.php?a=inicio">Início</a> |

<a href="index.php?a=servicos">Serviços</a> |

<a href="index.php?a=contatos">Contatos</a> |

<a href="index.php?a=mapa">Mapa</a> |

<a href="index.php?a=links">Links</a>

</div>

<hr />

**aula\_10\\_footer.php**

<!-- rodapé -->

<div>

<hr />

<p>Empresa @copy 2021</p>

<p><a href="mailto: empresa@gmail.com">empresa@gmail.com</a></p>

</div>

</body>

</html>

**aula\_10\index.php**

<?php

// inclui o cabeçalho

include('\_header.php');

// analisar que conteúdo apresentar

$acao = 'inicio';

// verifica se 'a' está definido

if(isset($\_GET['a'])){

$acao = $\_GET['a'];

}

include('caminhos.php');

// inclui o rodapé

include('\_footer.php');

?>

**aula\_10\caminhos.php**

<?php

// análise da ação

switch($acao){

case 'inicio':

include 'inicio.php';

break;

case 'servicos':

include 'servicos.php';

break;

case 'contatos':

include 'contatos.php';

break;

case 'links':

include 'links.php';

break;

case 'mapa':

include 'mapa.php';

break;

}

?>

**aula\_10\inicio.php**

<!-- Conteúdo -->

<div>

<h1>Início</h1>

</div>

**aula\_10\servicos.php**

<!-- Conteúdo -->

<div>

<h1>Serviços</h1>

</div>

**aula\_10\contatos.php**

<!-- Conteúdo -->

<div>

<h1>Contatos</h1>

</div>

**aula\_10\links.php**

<!-- Conteúdo -->

<div>

<h1>Links úteis</h1>

</div>

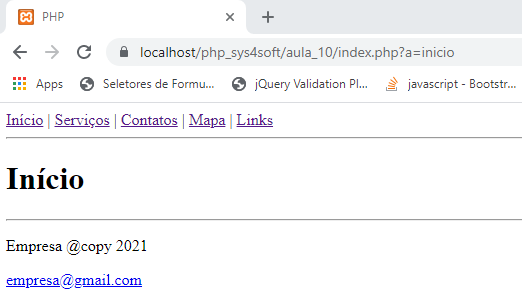
**aula\_10\mapa.php**

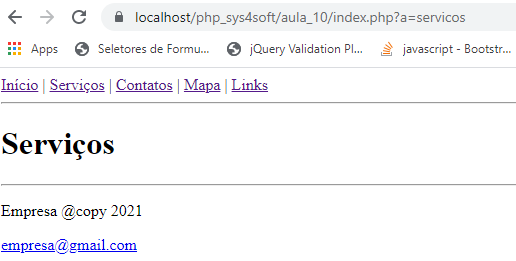
<!-- Conteúdo -->

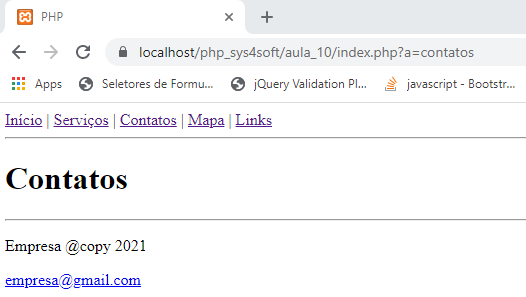
<div>

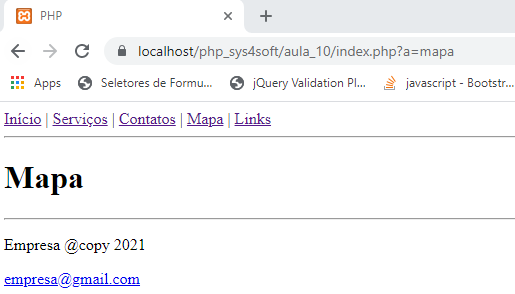
<h1>Mapa</h1>

</div>



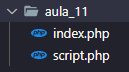








# Aula 11 - include vs require



## Sem erros

- Quando não há erros, o comportamento de ambos é o mesmo.

**aula\_11\index.php**

<?php

echo '<h1>Include vs Require</h1>';

$metodo = 'include';

echo "<p>Método usado: <strong>$metodo</strong>.</p>";

include 'script.php';

// conteúdo restante do index.php

echo '<p>Este é o conteúdo restante de index.php.</p>';

echo '================================================';

$metodo = 'require';

echo "<p>Método usado: <strong>$metodo</strong>.</p>";

require 'script.php';

// conteúdo restante do index.php

echo '<p>Este é o conteúdo restante de index.php.</p>';

?>

**aula\_11\script.php**

<?php

echo 'Frase contida em script.php';

?>

- No browser:



## Quando há erros

No caso do include, quando há erro, ele exibe a mensagem de erro ou warning e continua a execução do programa.

Já no caso do require, quando há erro, ele exibe a mensagem de erro ou warning e interrompe a execução de todas as linhas que vem a seguir.

**aula\_11\index.php**

<?php

echo '<h1>Include vs Require</h1>';

$metodo = 'include';

echo "<p>Método usado: <strong>$metodo</strong>.</p>";

include 'script1.php';

// conteúdo restante do index.php

echo '<p>Este é o conteúdo restante de index.php.</p>';

echo '================================================';

$metodo = 'require';

echo "<p>Método usado: <strong>$metodo</strong>.</p>";

require 'script1.php';

// conteúdo restante do index.php

echo '<p>Este é o conteúdo restante de index.php.</p>';

?>

**aula\_11\script.php**

<?php

echo 'Frase contida em script.php';

?>



# Aula 12 - Namespaces no PHP

**aula\_12\index.php**

<?php

function falar(){

echo 'Falei';

}

require 'amigo.php';

FuncoesAmigo\falar();

?>

**aula\_12\amigo.php**

<?php

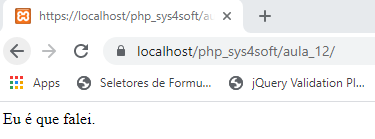
namespace FuncoesAmigo;

function falar(){

echo 'Eu é que falei.';

}

?>



# Aula 13 - Não podemos esquecer os arrays

## Array

**aula\_13\array.php**

<?php

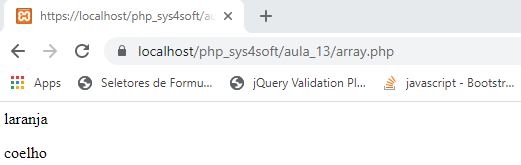
$frutas = array("laranja", "banana", "pera", "morango");

echo "<p>$frutas[0]</p>";

$animais = ["gato", "cachorro", "cavalo", "coelho"];

echo "<p>$animais[3]</p>";

?>



## Array associativo

**aula\_13\array\_associativo.php**

<?php

$frutas = array(10 => "laranja", 20 => "banana", 30 => "pera", 40 => "morango");

echo "<p>$frutas[20]</p>";

$animais = [

"felino" => "gato",

10 => "cachorro",

"cavalo",

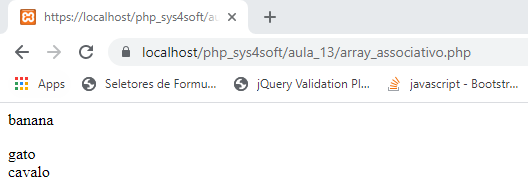
30 => "coelho"

];

echo $animais["felino"] . "<br />";

echo $animais[11] . "<br />";

?>



# Aula 14 - Programação orientada a objetos no PHP

**aula\_14\Testes.php**

<?php

class Testes {

public $texto = "Texto";

public function Criar() : void{

echo 'Método da classe' . "<br />";

}

public static function Falar($mensagem){

echo 'Eu falei ' . $mensagem . "<br />";

}

}

$a = new Testes();

$a->Criar();

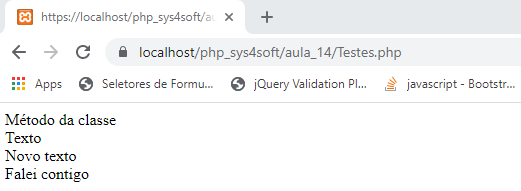
echo $a->texto . "<br />";

$a->texto = "Novo texto";

echo $a->texto . "<br />";

echo Testes::Falar('contigo');

?>



# Aula 15 - Ciclo de vida de um objeto

**aula\_15\Humano.php**

<?php

class Humano {

function \_\_construct(){

echo 'Um humano nasceu <br />';

}

public function Falar($texto){

echo 'O humano disse: ' . $texto . "<br />";

}

function \_\_destruct(){

echo 'O humano morreu <br />';

}

}

$homem1 = new Humano();

$homem1->Falar('estou vivo!');

?>



# Aula 16 - Hereditariedade de classes

**aula\_16\index.php**

<?php

abstract class Humano {

public $nome;

public $idade;

function \_\_construct($nome, $idade){

$this->nome = $nome;

$this->idade = $idade;

}

public function Falar($mensagem){

echo "<p>$mensagem</p>";

}

public function Apresentar(){

echo "<p>Meu nome é $this->nome, tenho $this->idade anos de idade!</p>";

}

}

class Homem extends Humano {

public function OQueEstaFazendo(){

echo "<p>Estou fazendo a barba!</p>";

}

}

class Mulher extends Humano {

public function OQueEstaFazendo(){

echo "<p>Estou me maquiando!</p>";

}

}

$homem = new Homem('João', 25);

$homem->Falar('Bom dia!');

$homem->Apresentar();

$homem->OQueEstaFazendo();

echo "=====================================";

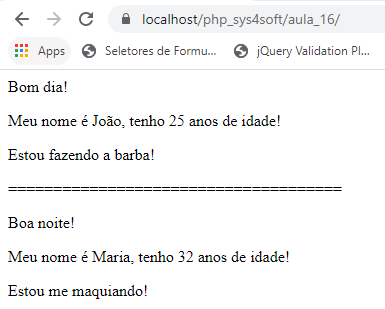
$mulher = new Mulher('Maria', 32);

$mulher->Falar('Boa noite!');

$mulher->Apresentar();

$mulher->OQueEstaFazendo();

?>



# Aula 17 - Operadores de comparação

**aula\_17\index.php**

<?php

compara(15, 15);

compara(15, '15');

compara(5, '15');

function compara($a, $b){

if($a == $b){

if ($a === $b){

echo "<p>Possuem mesmo valor e tipo</p>";

} else {

echo "<p>Possuem mesmo valor, porém tipos diferentes</p>";

echo '<p>tipo $a = ' . gettype($a) . '</p>';

echo '<p>tipo $b = ' . gettype($b) . '</p>';

}

} else {

echo '<p>São diferentes!</p>';

}

echo "===========================================";

}

$c = "";

$d = null;

if(empty($c)){

echo '<p>A variável $c está vazia</p>';

}

if(is\_null($d)){

echo '<p>A variável $d é nula</p>';

}

echo "===========================================";

if(isset($c)){

echo '<p>A variável $c contém um valor válido</p>';

} else {

echo '<p>A variável $c contém um valor inválido</p>';

}

if(isset($d)){

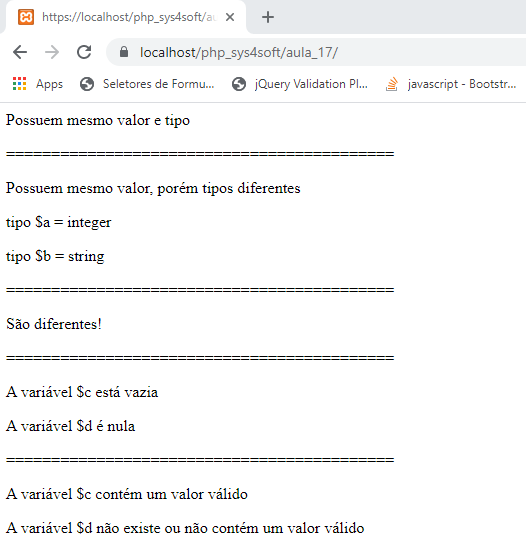
echo '<p>A variável $d contém um valor válido</p>';

} else {

echo '<p>A variável $d não existe ou não contém um valor válido</p>';

}

?>



# Aula 18 - Aspectos relacionados com instruções condicionais

- Tipo e valor podem ser analisados com o if, mas não são analisados com o switch.

**aula\_18\index.php**

<?php

function teste($nome) {

if(gettype($nome) === "string")

return "<p>É uma string</p>";

return "<p>Não é uma string</p>";

}

echo teste('Maria');

echo teste(5);

$a = '2';

switch($a){

case 1:

echo '<p>Valor 1</p>';

break;

case 2:

echo '<p>Valor 2</p>';

break;

case 3:

echo '<p>Valor 3</p>';

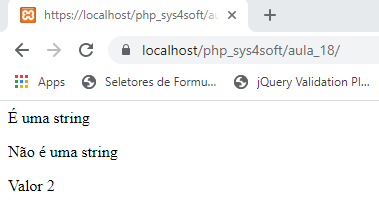
break;

default:

echo '<p>Outro valor</p>';

}

?>



# Aula 19 - Atenção especial ao controle de erros

**aula\_19\erros\_1.php**

<?php

if(!file\_exists('teste.php')) {

die('<p>O arquivo não existe!</p>');

} else {

include 'teste.php';

}

?>



**aula\_19\erros\_2.php**

<?php

$valor = 1;

if(isset($valor)){

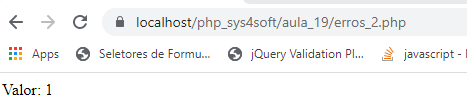
echo "<p>Valor: $valor</p>";

} else {

echo '<p>A variável não está definida.</p>';

}

?>



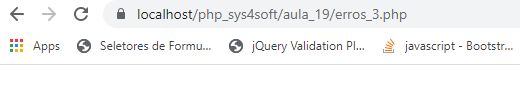
**aula\_19\erros\_3.php**

<?php

error\_reporting(0); // não exibe os erros

require 'teste.php';

?>



**aula\_19\erros\_4.php**

<?php

error\_reporting(E\_WARNING); // Exibe apenas as mensagens do tipo warning

require 'teste.php';

?>



## Lançando uma exceção (throw)

**aula\_19\erros\_5.php**

<?php

// create function with an exception

function checkNum($number) {

if($number > 1) {

throw new Exception("Valor deve ser menor ou igual a 1");

}

return true;

}

// trigger exception

try{

$result = checkNum(2);

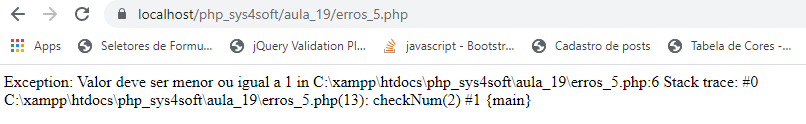
echo $result;

} catch(Exception $e) {

echo $e;

}

?>



# Aula 20 - DateTime no PHP

**aula\_20\index.php**

<?php

// Exibindo datas

$data\_atual = date('d/m/Y H:i:s');

echo "<p>$data\_atual</p>";

echo "===========================================";

echo "<h3>Exibindo datas </h3>";

$data\_ontem = new DateTime('yesterday');

echo "<p>Ontem: " . $data\_ontem->format('d/m/Y') . "</p>";

$data\_hoje = new DateTime('today');

echo "<p>Hoje: " . $data\_hoje->format('d/m/Y') . "</p>";

$data\_amanha = new DateTime('tomorrow');

echo "<p>Amanhã: " . $data\_amanha->format('d/m/Y') . "</p>";

echo "===========================================";

// Comparando datas

echo "<h3>Comparando datas</h3>";

if($data\_ontem < $data\_hoje){

echo '<p>$data\_ontem é menor que $data\_hoje</p>';

}

if($data\_amanha > $data\_hoje) {

echo '<p>$data\_amanha é maior que $data\_hoje</p>';

}

echo "===========================================";

// Adicionando dias a uma data

echo "<h3>Adicionando 10 dias a data de hoje</h3>";

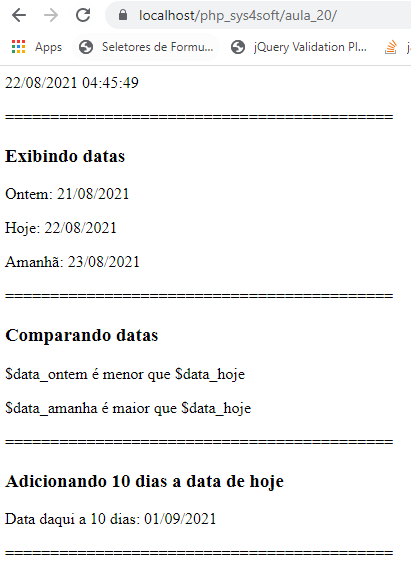
$data\_plus\_10\_dias = $data\_hoje->add(new DateInterval('P10D'));

// Hoje + 10 dias

echo "<p>Data daqui a 10 dias: " . $data\_plus\_10\_dias->format('d/m/Y') . "</p>";

echo "===========================================";

?>



# Aula 21 - Exemplo de uma classe para tratamento de datas e horas

**aula\_21\DataHora.php**

<?php

// =================================

// Classe para gerir datas e horas

// =================================

class DataHora {

public $data;

// --------------------------------------------

// construtor da classe

public function \_\_construct($formato = null){

// timezone

date\_default\_timezone\_set('America/Sao\_Paulo');

if($formato === null){

$this->data = new DateTime();

} else {

$this->data = new DateTime($formato);

}

}

// --------------------------------------------

public function getData($formato = null){

if($formato === null) {

return "<p>" . $this->data->format('d/m/Y') . "</p>";

} else {

return "<p>" . $this->data->format($formato) . "</p>";

}

}

public function getDataHora(){

return "<p>" . $this->data->format('d/m/Y H:i:s') . "</p>";

}

// --------------------------------------------

// Adicionar elementos a data

// --------------------------------------------

public function addDias($n){

$this->data->add(new DateInterval('P'.$n.'D'));

}

// --------------------------------------------

public function addMeses($n){

$this->data->add(new DateInterval('P'.$n.'M'));

}

// --------------------------------------------

public function addAnos($n){

$this->data->add(new DateInterval('P'.$n.'Y'));

}

// --------------------------------------------

// Adicionar elementos a hora

// --------------------------------------------

public function addHoras($n){

$this->data->add(new DateInterval('PT'.$n.'H'));

}

// --------------------------------------------

public function addMinutos($n){

$this->data->add(new DateInterval('PT'.$n.'I'));

}

// --------------------------------------------

public function addSegundos($n){

$this->data->add(new DateInterval('PT'.$n.'S'));

}

}

?>

**aula\_21\index.php**

<?php

require 'DataHora.php';

$d1 = new DataHora();

$d2 = new DataHora();

$d3 = new DataHora();

echo "<p>Data de Hoje</p>";

echo $d1->getData();

echo $d1->getData('Y-m-d');

echo "============================";

// adicionar 100 dias

echo "<p>Adicionando 100 dias a data de hoje</p>";

echo $d2->getData();

$d2->addDias(100);

echo $d2->getData();

echo "=============================";

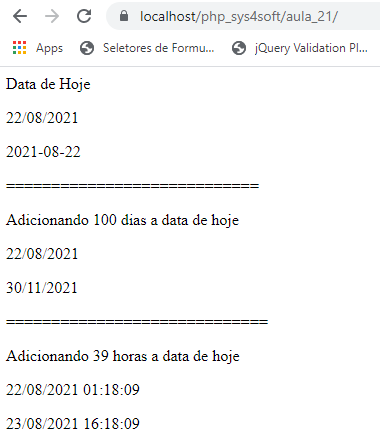
echo "<p>Adicionando 39 horas a data de hoje</p>";

echo $d3->getDataHora();

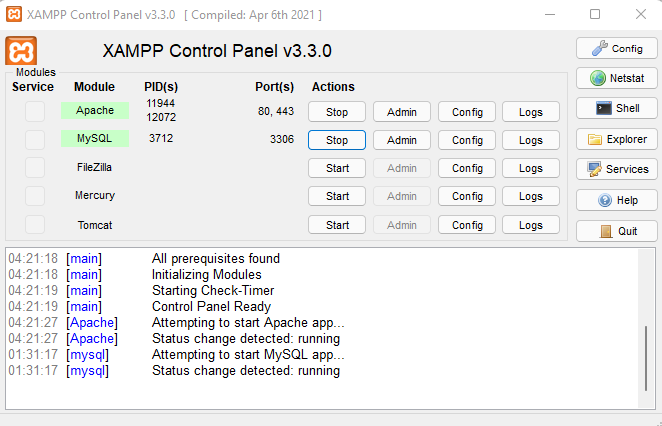
$d3->addHoras(39);

echo $d3->getDataHora();

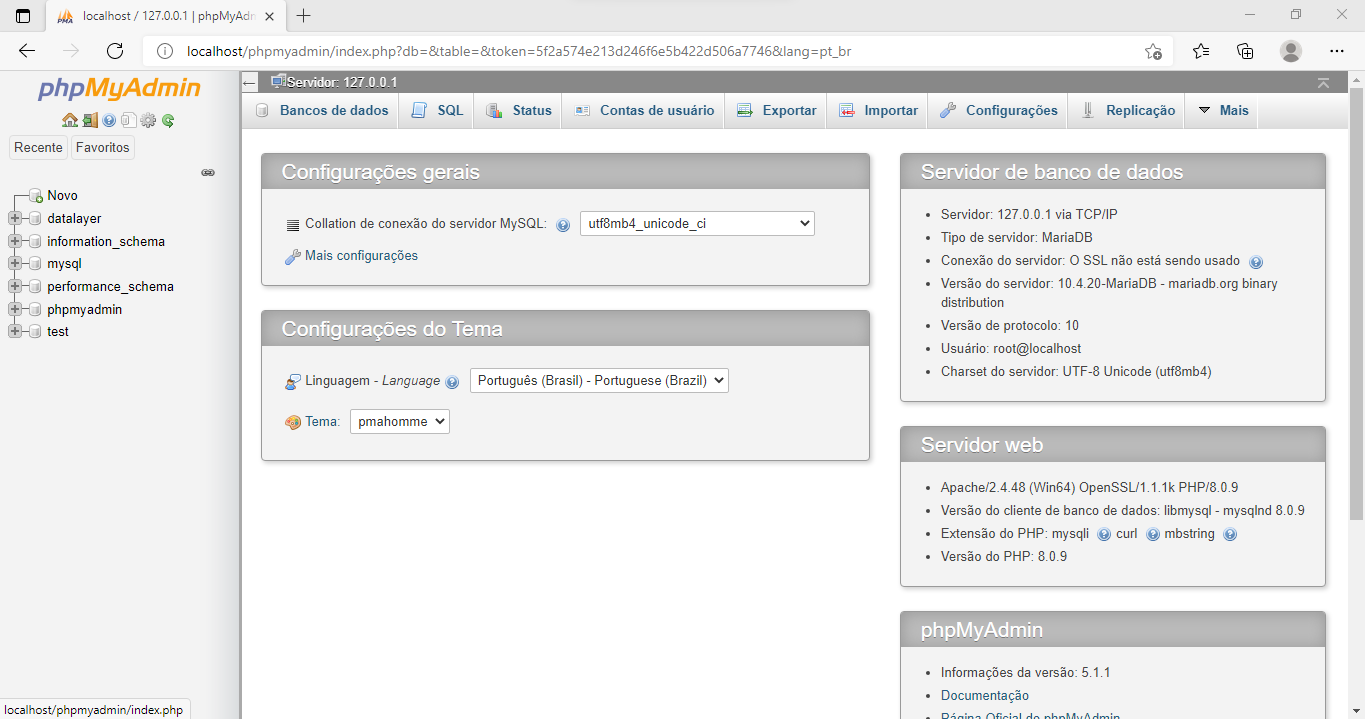
?>



# Aula 22 - PHP e MySQL através de PDO - PHPMyAdmin

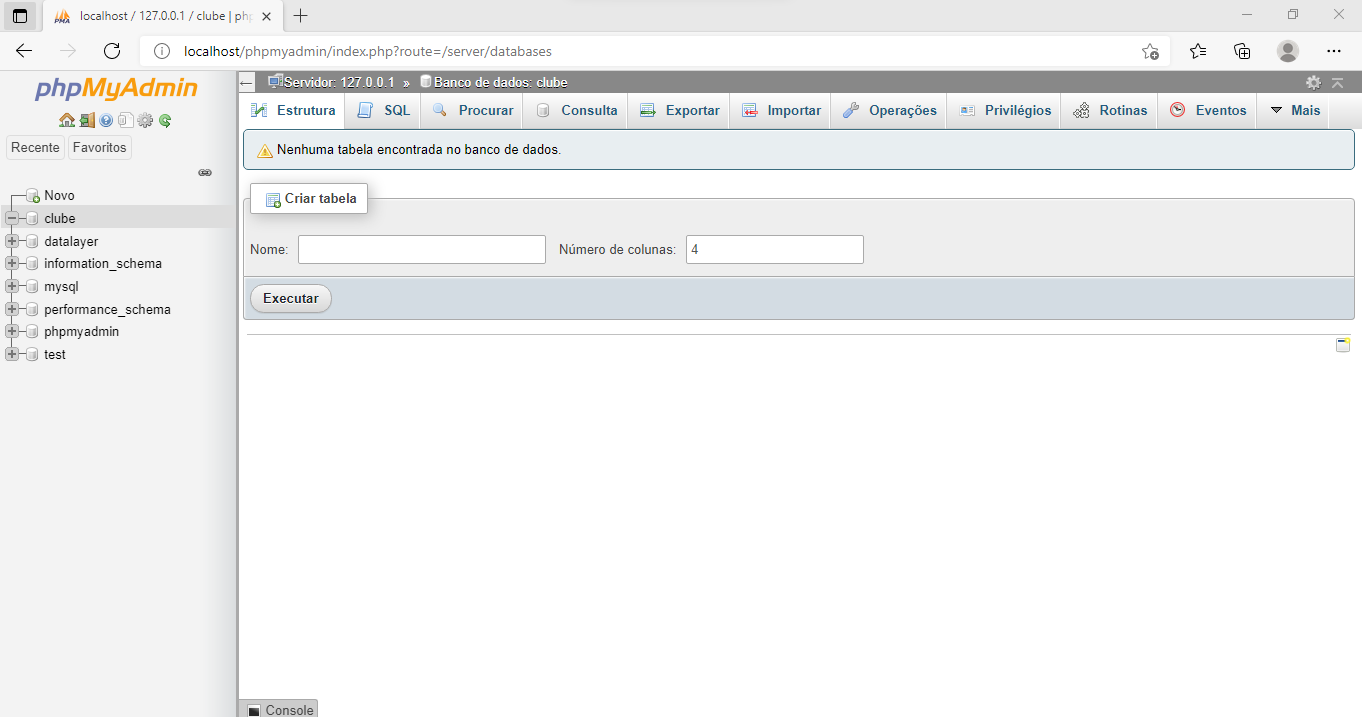


- No módulo MySQL, clique no botão "Admin":



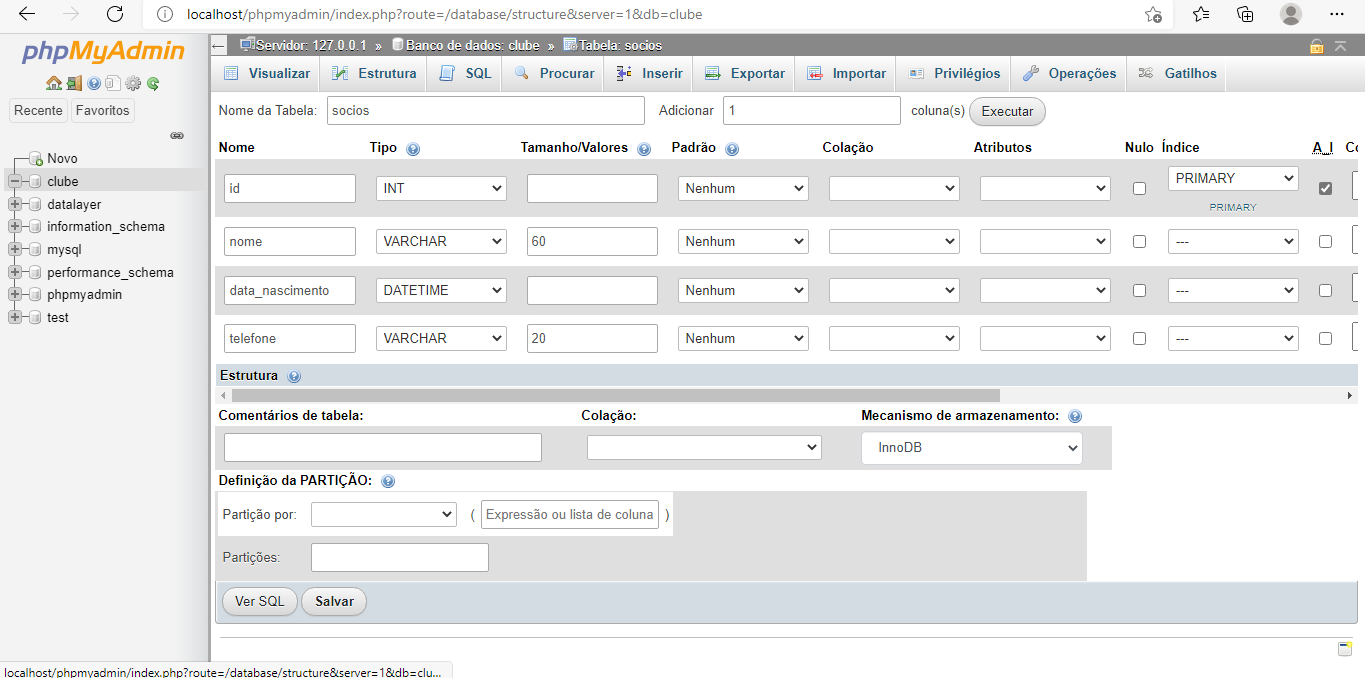
## Criando uma base de dados

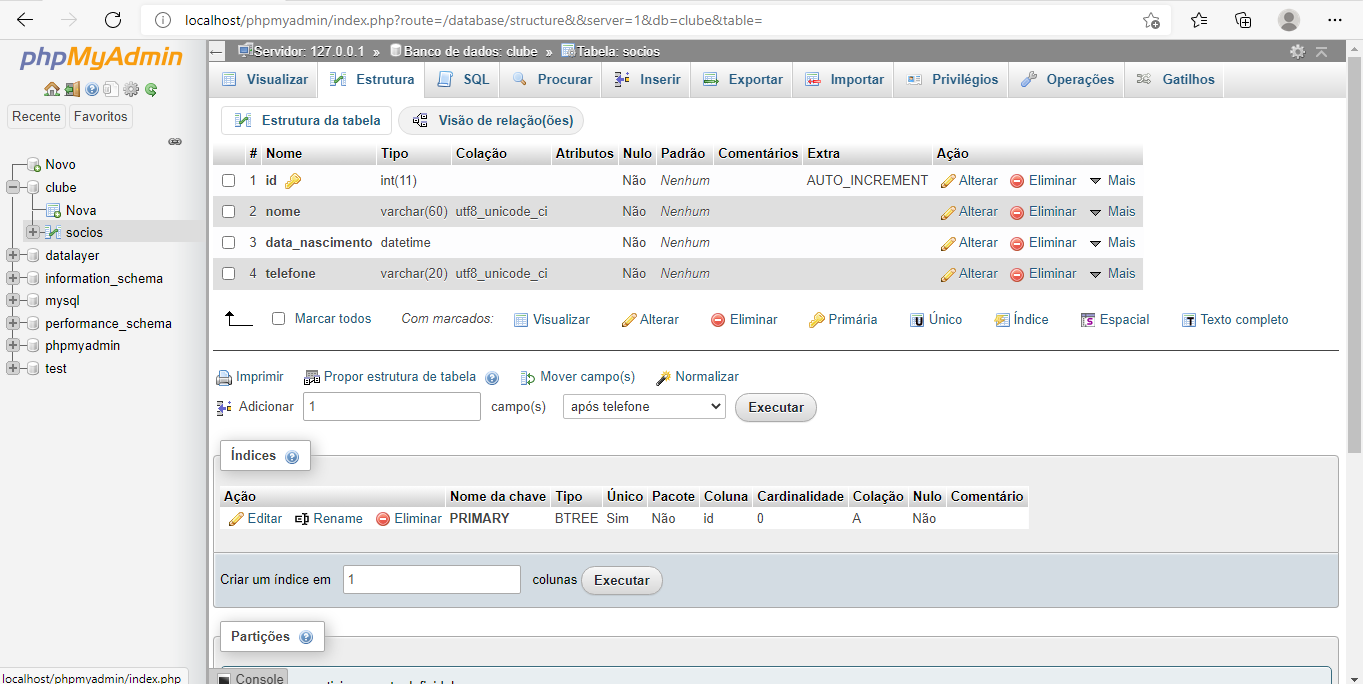




# Aula 23 - PHP e MySQL - PHPMyAdmin - criar uma base de dados

- No BD clube, crie a tabela sócios com a estrutura a seguir:



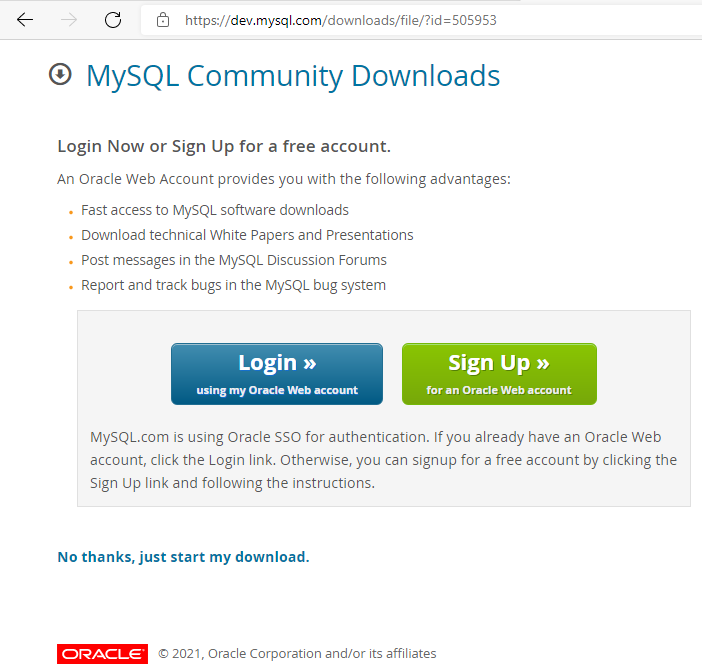


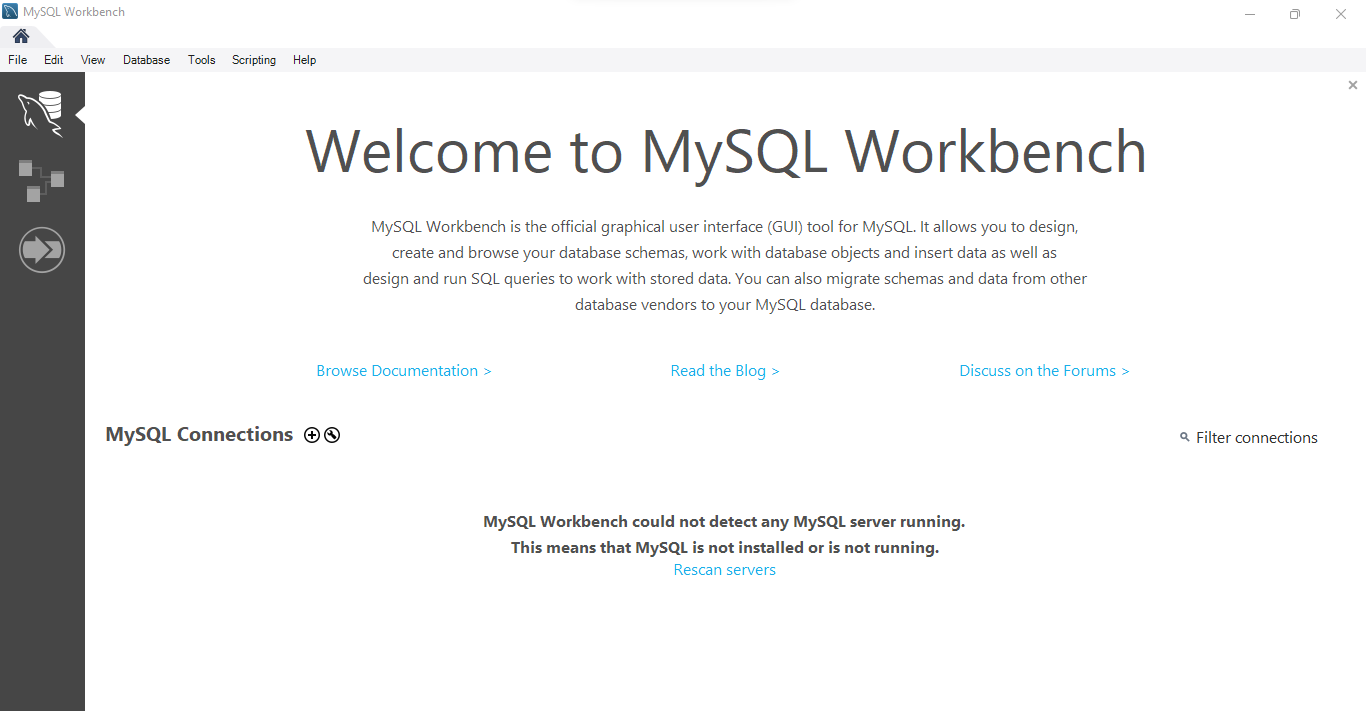
# Aula 24 - PHP e MySQL - Alternativas ao PHPMyAdmin

## MySQL Workbench

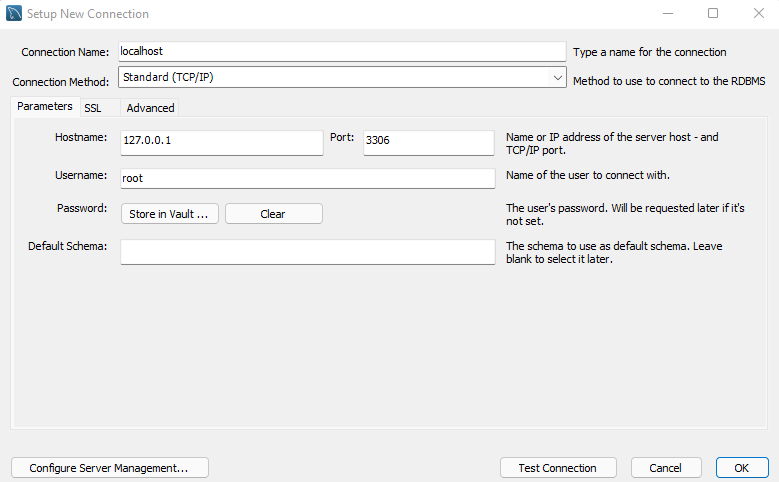
https://dev.mysql.com/downloads/workbench/

- Baixe e instale o MySQL Workbench.

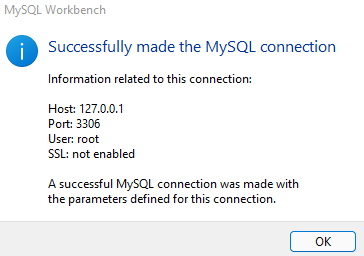


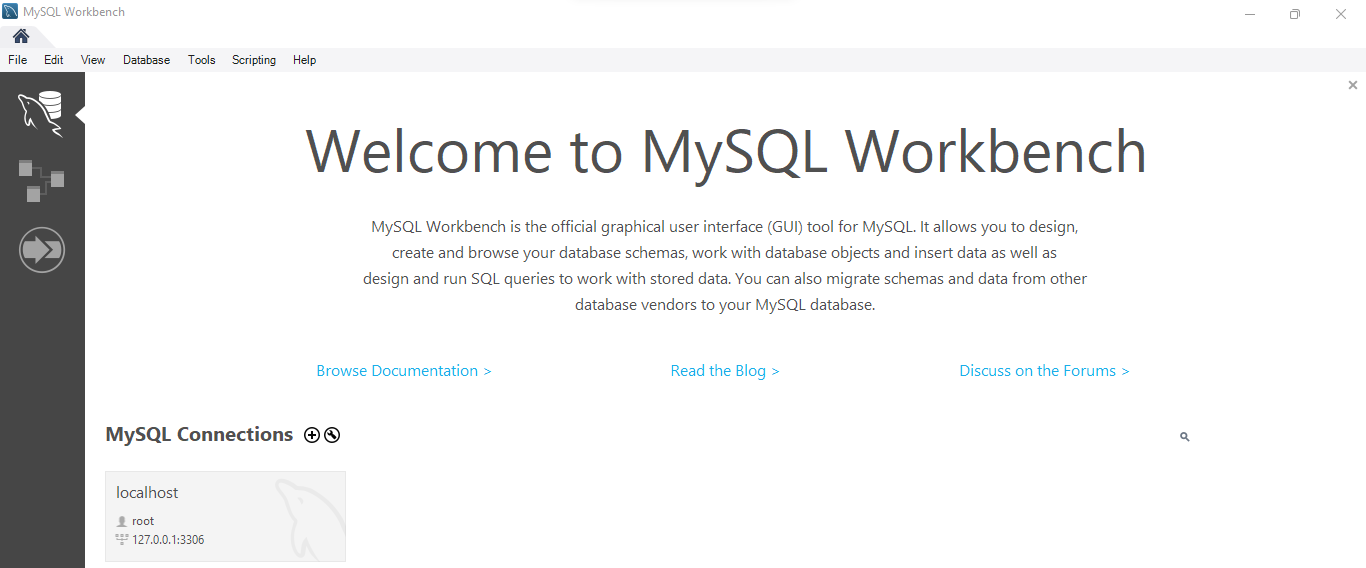


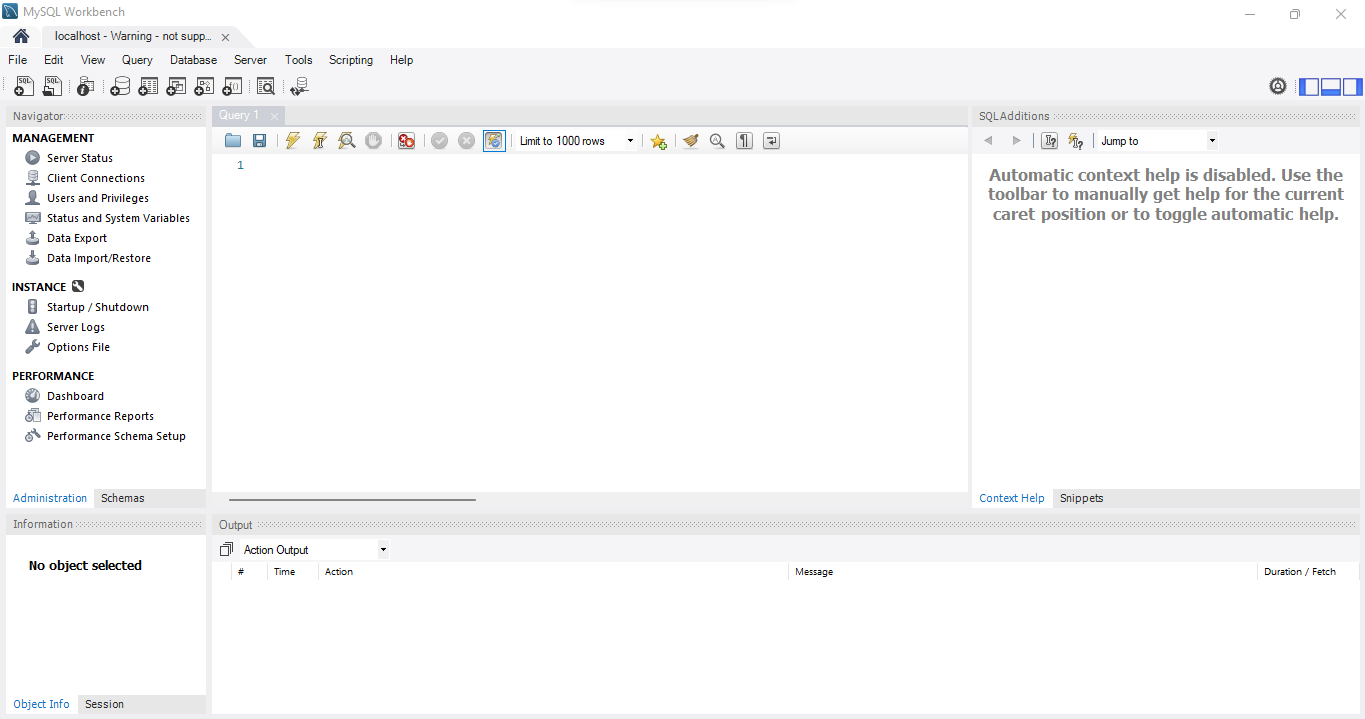
- Adicione uma nova conexão:



- Clique no botão Test Connection:

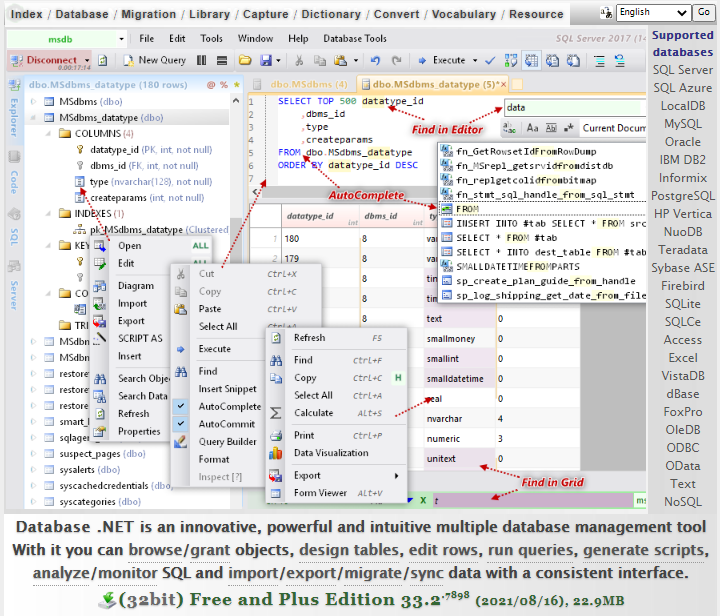


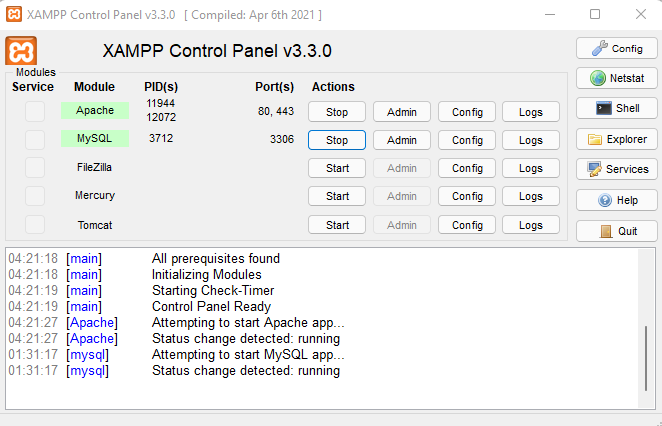


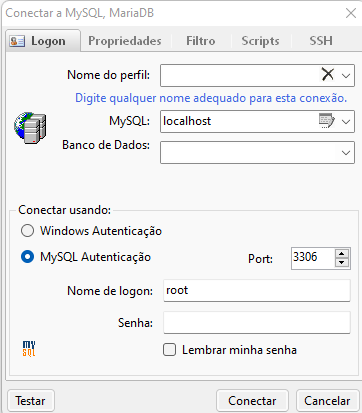


## Database .NET

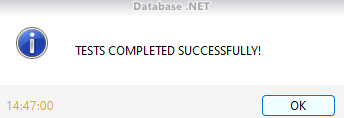
https://fishcodelib.com/Database.htm



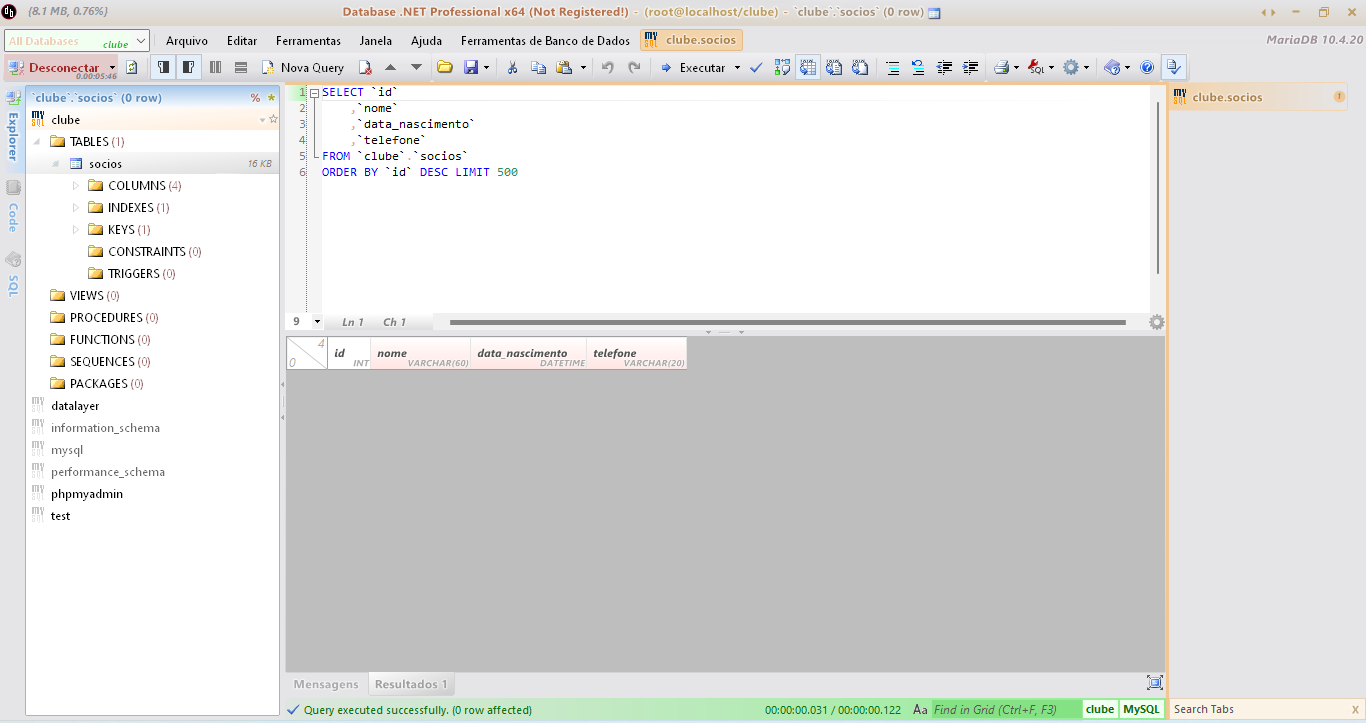


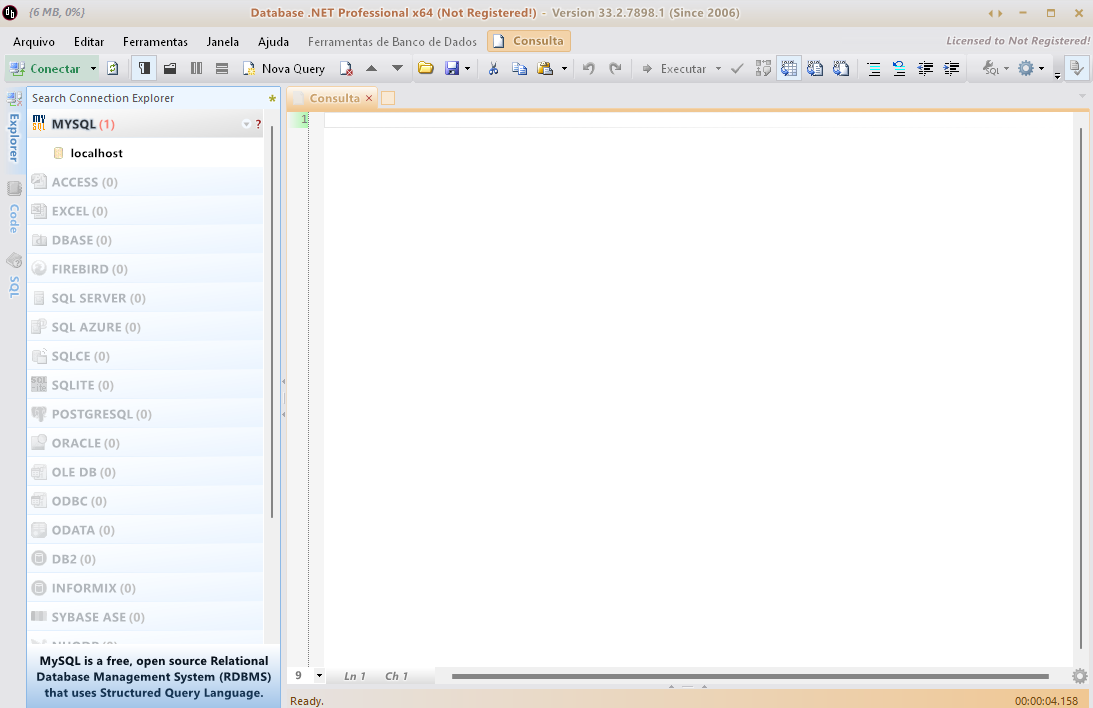


- Clique no botão "Testar":



- Clique no botão "OK" e em seguida no botão "Conectar":



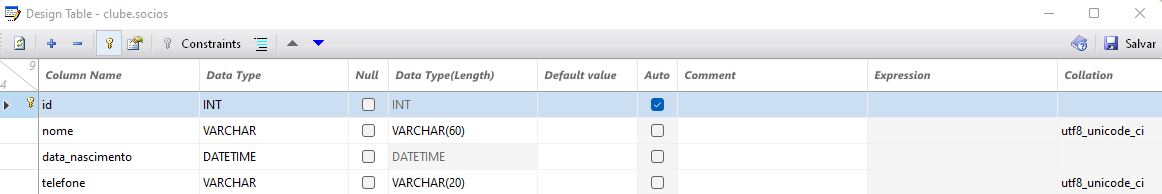


## Outras alternativas

* Navicat for mySQL (aplicação comercial - não gratuita)
* Sequel Pro
* HeidiSQL
* SQLyog Community Edition
* SQLWave

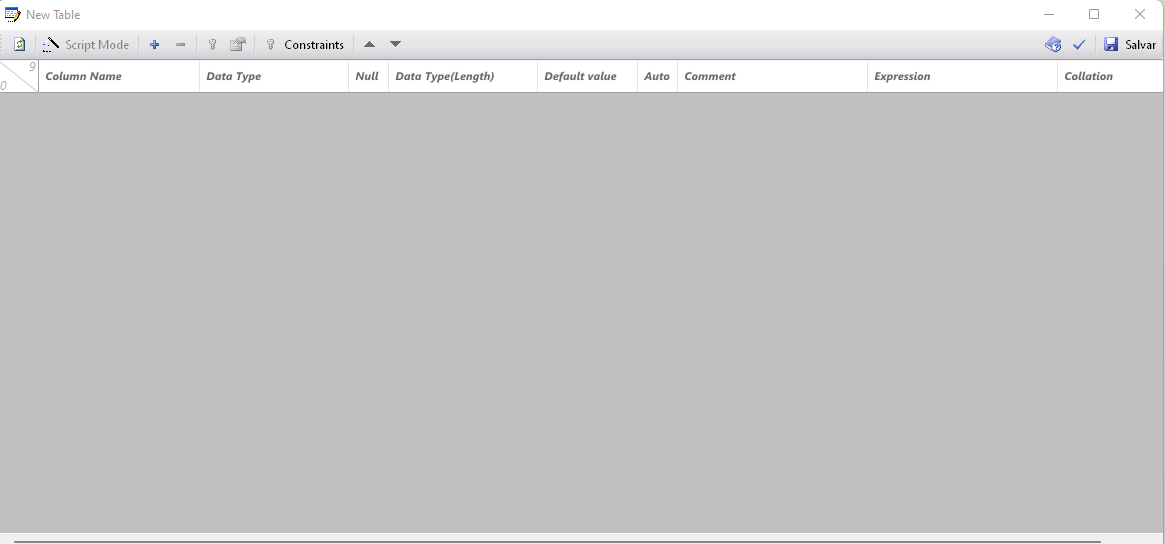
# Aula 26 - Criação de BD MySQL com database .NET

- Para ver e/ou alterar a estrutura de dados, clique em "Design":

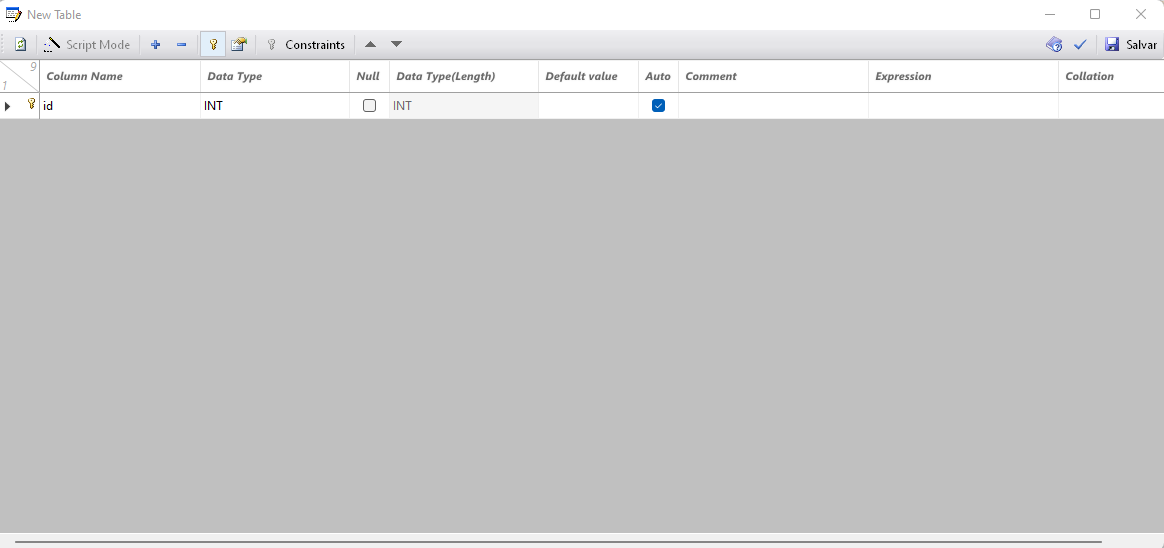


## Criando uma tabela

TABLES --> CREATE TABLE

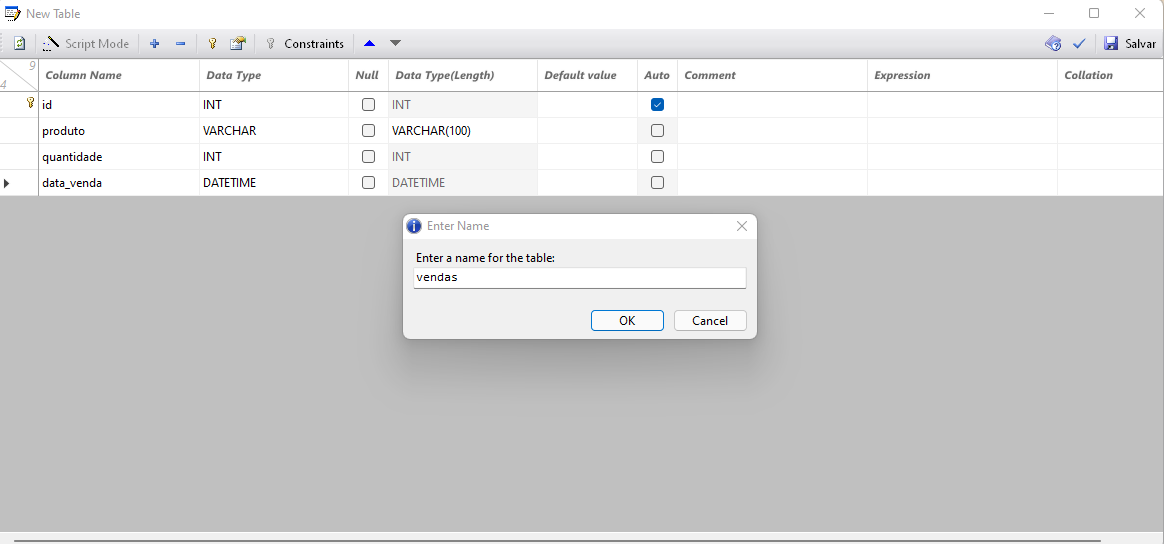


- Clique no botão + e insira o nome da coluna:

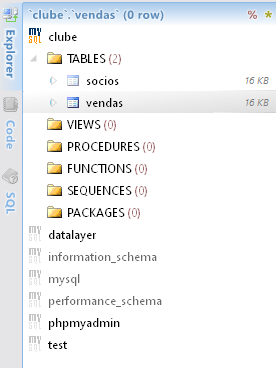


- Clique no botão Set Primary Key.

- Clicando no botão + adicione novas colunas:



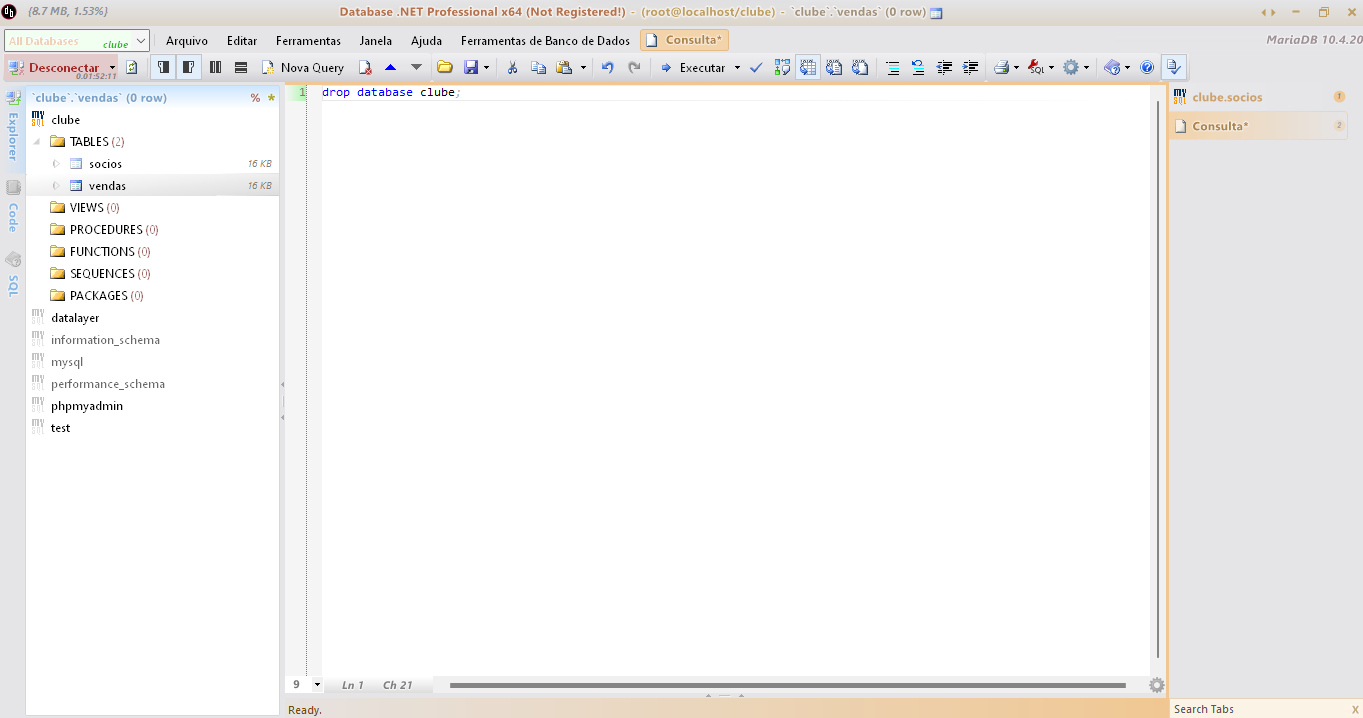
- Salve a tabela com o nome de "vendas".



# Aula 27 - Integridade referencial e auto incremento

## Apagando a base de dados "clube"

drop database clube;

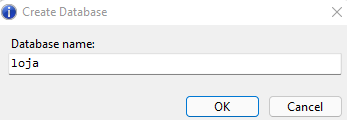


- Clique no botão "Executar".

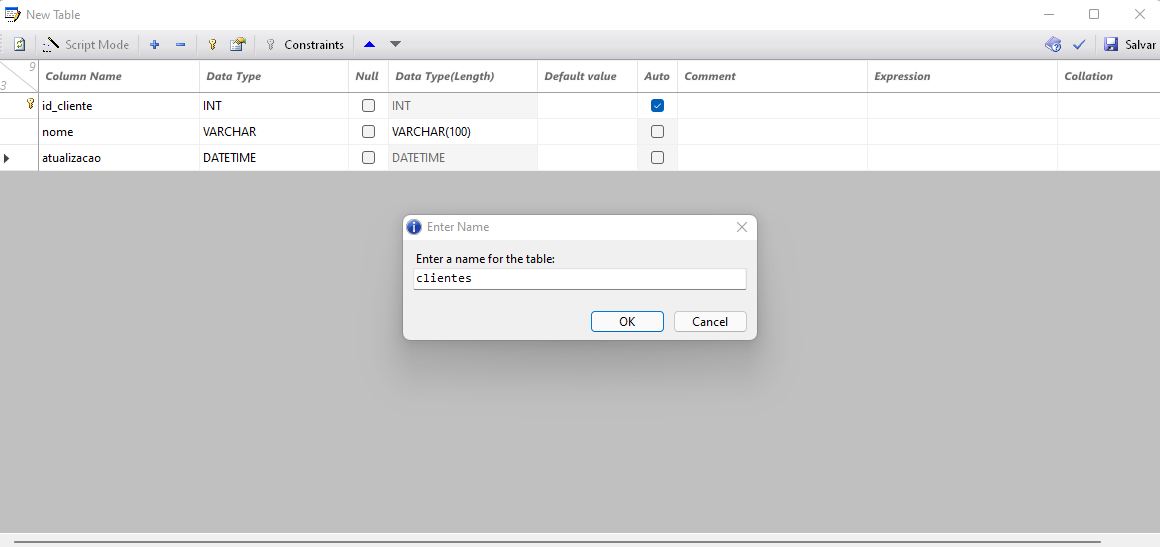
## Criando uma nova base de dados

- Crie uma nova base de dados chamada "loja" com duas tabelas "clientes" e "vendas".

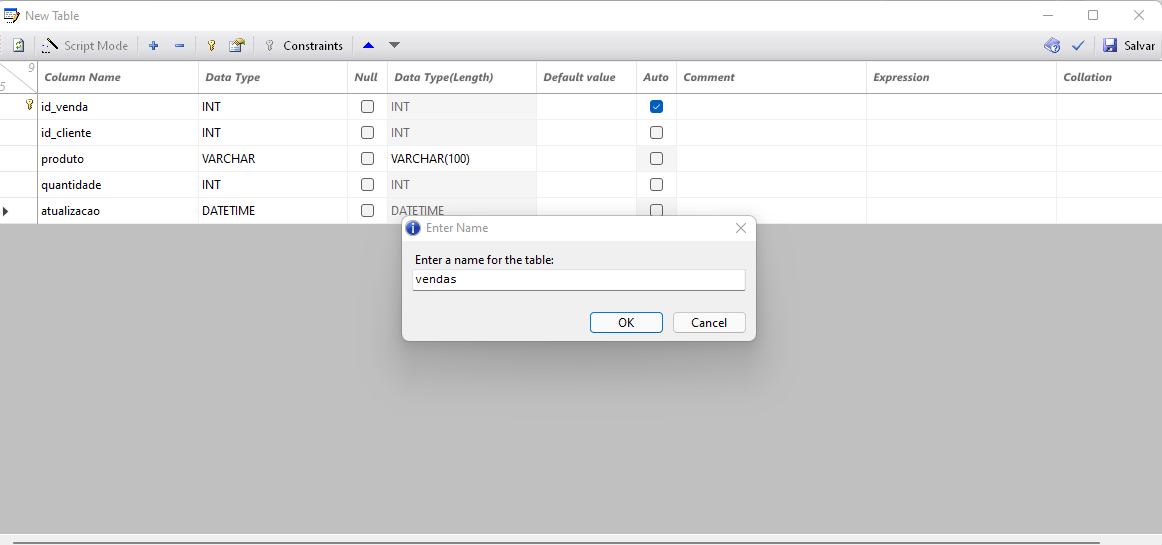
Integridade referencial: quando se exclui um cliente todas as vendas a este cliente também devem ser automaticamente excluídas.



- Crie a tabela "clientes":



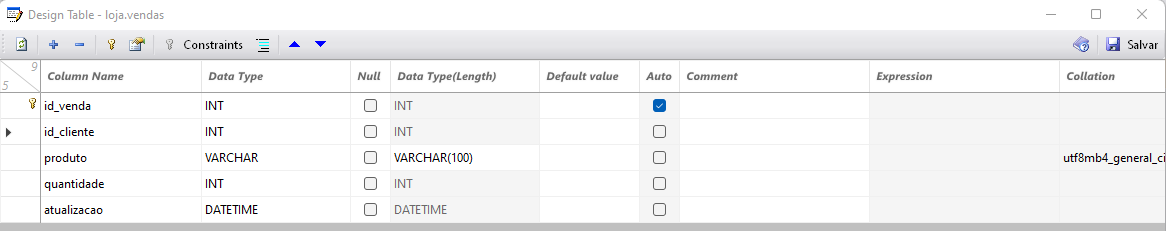
- Crie a tabela "vendas":



- Na tabela "vendas" a coluna "id\_cliente" deve ser uma chave estrangeira (foreign key).

## Criando a chave estrangeira

vendas --> design

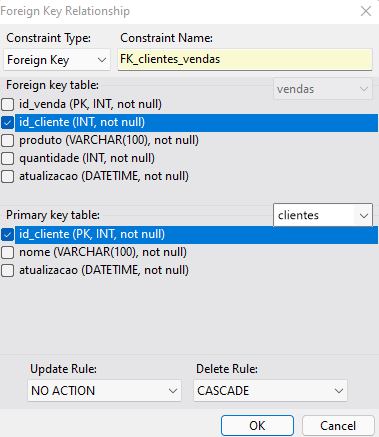


- Selecione o campo "id\_cliente":

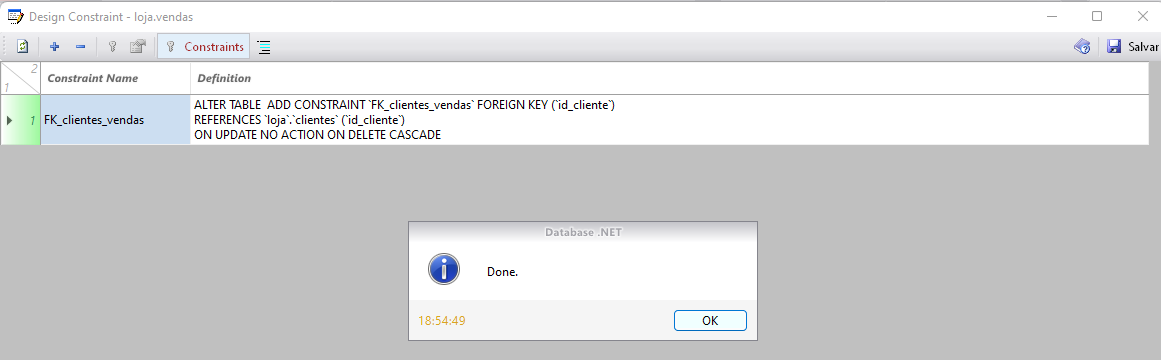
- Dê um clique no botão "Constraints":



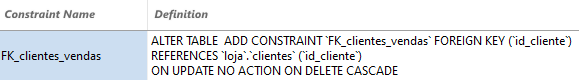
- Clique no botão "+" e preencha o formulário aberto da seguinte forma:



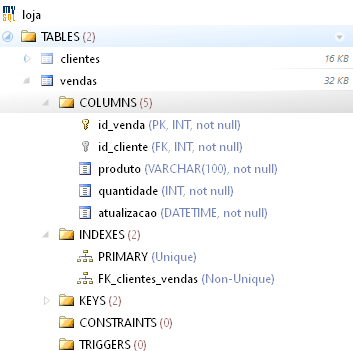
- Clique no botão "OK":



- A query SQL da criação dessa chave estrangeira é:

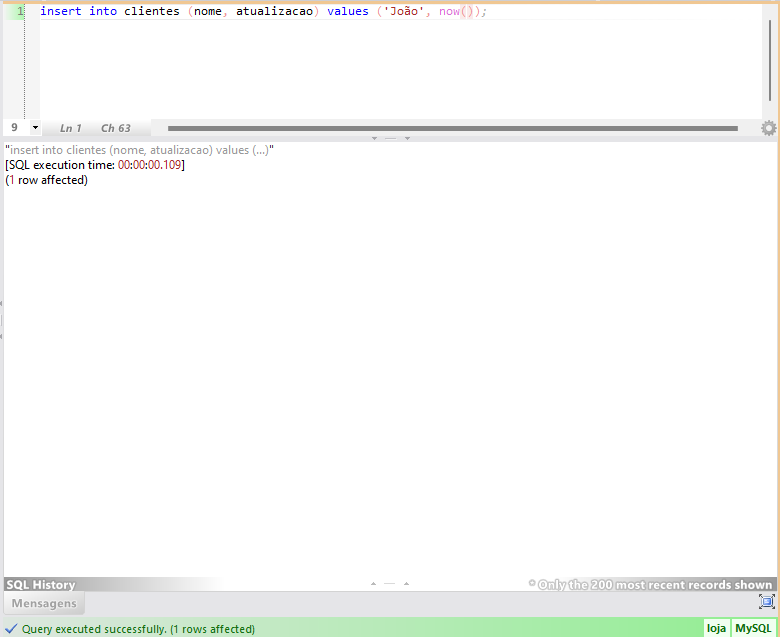


ALTER TABLE ADD CONSTRAINT 'FK\_clientes\_vendas' FOREIGN KEY ('id\_cliente') REFERENCES 'loja'.'clientes'('id\_cliente') ON UPDATE NO ACTION ON DELETE CASCADE

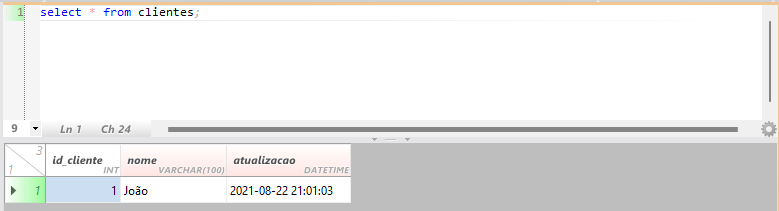


## Inserindo um cliente

insert into clientes (nome, atualizacao) values ('João', now());



select \* from clientes;



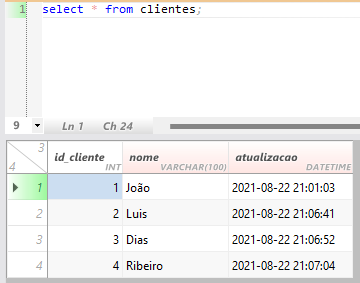
## Inserindo outros registros

insert into clientes (nome, atualizacao) values ('Luis', now());

insert into clientes (nome, atualizacao) values ('Dias', now());

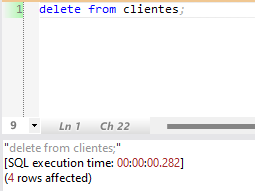
insert into clientes (nome, atualizacao) values ('Ribeiro', now());

select \* from clientes;



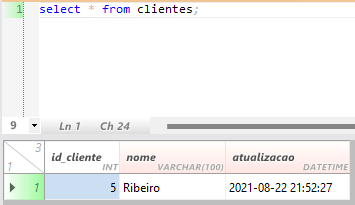
## Apagando todos os clientes

delete from clientes;



- Insira um novo cliente:

insert into clientes (nome, atualizacao) values ('João', now());

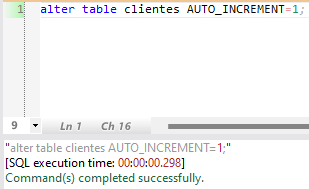


- Perceba que o número de id é 5 e não 1.

## Resetando o id

- Apague o registro que foi inserido e entre com o seguinte comando SQL:

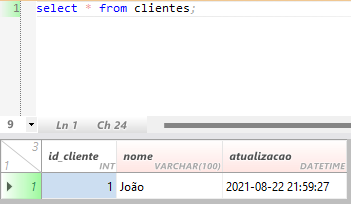
alter table clientes AUTO\_INCREMENT=1;



- Insira novamente o cliente:

insert into clientes (nome, atualizacao) values ('João', now());

select \* from clientes;



# Aula 28 - Acesso a base de dados com PDO

## Criando conexão com a base de dados usando PDO

**aula\_28\index.php**

<?php

$host = "localhost";

$dbname = "loja";

$dbusername = "root";

$dbpassword = "";

// $gestor = new PDO(ligacao, username, password);

$pdo = new PDO("mysql:host=$host; dbname=$dbname", $dbusername, $dbpassword);

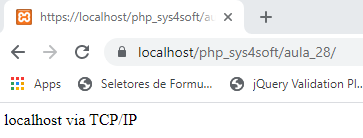
$estado = $pdo->getAttribute(PDO::ATTR\_CONNECTION\_STATUS);

echo $estado;

?>

- No browser:

localhost/php\_sys4soft/aula\_28



# Aula 29 - Criação de uma classe PDO - parte1

**aula\_29\index.php**

<?php

require 'config.php';

require 'gestor.php';

$a = new gestor();

$a->EXEC\_QUERY("");

?>

**aula\_29\config.php**

<?php

// ======================================

// Variáveis de configuração do sistema

// ======================================

// base de dados

$db\_host = "localhost";

$db\_name = "loja";

$db\_charset = "utf8";

$db\_username = "root";

$db\_password = "";

?>

**aula\_29\gestor.php**

<?php

// classe para a gestão da BD

class gestor {

// ===============================================

public function EXE\_QUERY($query, $parametros = null){

// resultados

$resultados = null;

// abre a conexão à BD

$conn = new PDO('mysql:host='.$GLOBALS['db\_host'].

';bdname='.$GLOBALS['db\_name'].

';charset='.$GLOBALS['db\_charset'],

$GLOBALS['db\_username'],

$GLOBALS['db\_password']

);

echo $conn->getAttribute(PDO::ATTR\_CONNECTION\_STATUS);

return $resultados;

}

// ===============================================

public function EXE\_NON\_QUERY(){

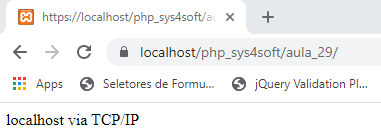
}

}

?>

- No browser:

https://localhost/php\_sys4soft/aula\_29/



# Aula 30 - Criação de uma classe PDO - parte2

**aula\_30\config.php**

<?php

// ======================================

// Variáveis de configuração do sistema

// ======================================

// base de dados

$db\_host = "localhost";

$db\_name = "loja";

$db\_charset = "utf8";

$db\_username = "root";

$db\_password = "";

?>

**aula\_30\gestor.php**

<?php

// classe para a gestão da BD

class gestor {

// ===============================================

public function EXE\_QUERY($query, $parametros = null){

// resultados

$resultados = null;

// SELECT

// abre a conexão à BD

$conn = new PDO('mysql:host='.$GLOBALS['db\_host'].

';bdname='.$GLOBALS['db\_name'].

';charset='.$GLOBALS['db\_charset'],

$GLOBALS['db\_username'],

$GLOBALS['db\_password'],

array(PDO::ATTR\_PERSISTENT=>TRUE)

);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

// executa a query

if($parametros != null){

$gest = $conn->prepare($query);

$gest->execute($parametros);

$resultados = $gest->fetchAll(PDO::FETCH\_ASSOC); // retorna os dados através de um array associativo

} else {

$gest = $conn->prepare($query);

$gest->execute();

$resultados = $gest->fetchAll(PDO::FETCH\_ASSOC); // retorna os dados através de um array associativo

}

// fechar a conexão

$conn = null;

return $resultados;

}

// ===============================================

public function EXE\_NON\_QUERY($query, $parametros = null){

// INSERT, UPDATE, DELETE

// abre a conexão à BD

$conn = new PDO('mysql:host='.$GLOBALS['db\_host'].

';bdname='.$GLOBALS['db\_name'].

';charset='.$GLOBALS['db\_charset'],

$GLOBALS['db\_username'],

$GLOBALS['db\_password'],

array(PDO::ATTR\_PERSISTENT=>TRUE)

);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

// executa a query

$conn->beginTransaction();

try{

if($parametros != null){

$gest = $conn->prepare($query);

$gest->execute($parametros);

} else {

$gest = $conn->prepare($query);

$gest->execute();

}

$conn->commit();

} catch(PDOException $e){

echo '<p>' . $e . '</p>';

$conn->rollBack();

}

$conn = null;

}

}

?>

**aula\_30\index.php**

<?php

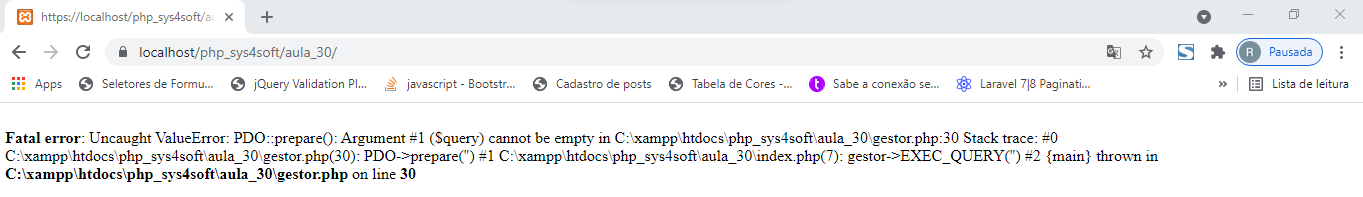
require 'config.php';

require 'gestor.php';

$a = new gestor();

$a->EXE\_QUERY("");

?>



# Aula 31 - Testes à criação da classe PDO

**aula\_31\config.php**

<?php

// ======================================

// Variáveis de configuração do sistema

// ======================================

// base de dados

$db\_host = "localhost";

$db\_name = "loja";

$db\_charset = "utf8";

$db\_username = "root";

$db\_password = "";

?>

**aula\_31\gestor.php**

<?php

// classe para a gestão da BD

class gestor {

// ===============================================

public function EXE\_QUERY($query, $parametros = null){

// resultados

$resultados = null;

// SELECT

// abre a conexão à BD

$conn = new PDO('mysql:host='.$GLOBALS['db\_host'].

';dbname='.$GLOBALS['db\_name'].

';charset='.$GLOBALS['db\_charset'],

$GLOBALS['db\_username'],

$GLOBALS['db\_password'],

array(PDO::ATTR\_PERSISTENT=>TRUE)

);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

// executa a query

if($parametros != null){

$gest = $conn->prepare($query);

$gest->execute($parametros);

$resultados = $gest->fetchAll(PDO::FETCH\_ASSOC); // retorna os dados através de um array associativo

} else {

$gest = $conn->prepare($query);

$gest->execute();

$resultados = $gest->fetchAll(PDO::FETCH\_ASSOC); // retorna os dados através de um array associativo

}

// fechar a conexão

$conn = null;

return $resultados;

}

// ===============================================

public function EXE\_NON\_QUERY($query, $parametros = null){

// INSERT, UPDATE, DELETE

// abre a conexão à BD

$conn = new PDO('mysql:host='.$GLOBALS['db\_host'].

';dbname='.$GLOBALS['db\_name'].

';charset='.$GLOBALS['db\_charset'],

$GLOBALS['db\_username'],

$GLOBALS['db\_password'],

array(PDO::ATTR\_PERSISTENT=>TRUE)

);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

// executa a query

$conn->beginTransaction();

try{

if($parametros != null){

$gest = $conn->prepare($query);

$gest->execute($parametros);

} else {

$gest = $conn->prepare($query);

$gest->execute();

}

$conn->commit();

} catch(PDOException $e){

echo '<p>' . $e . '</p>';

$conn->rollBack();

}

$conn = null;

}

}

?>

**aula\_31\index.php**

<?php

require 'config.php';

require 'gestor.php';

// Inserção do primeiro cliente

$gestor = new gestor();

$query = "insert into clientes (nome, atualizacao) values ('João', now())";

$gestor->EXE\_NON\_QUERY($query);

echo '<p>Cliente inserido com sucesso!</p>';

// Inserção do segundo cliente

$parametros = [

":nome" => 'Luis'

];

$query = "insert into clientes (nome, atualizacao) values (:nome, now())";

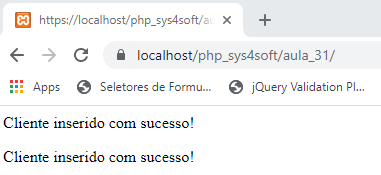
$gestor = new gestor();

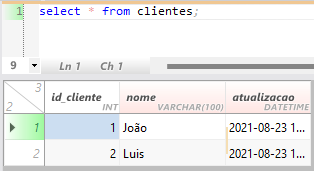
$gestor->EXE\_NON\_QUERY($query, $parametros);

echo '<p>Cliente inserido com sucesso!</p>';

?>

https://localhost/php\_sys4soft/aula\_31/





# Aula 32 - Leitura de dados com a classe PDO

**aula\_32\index1.php**

<?php

require 'config.php';

require 'gestor.php';

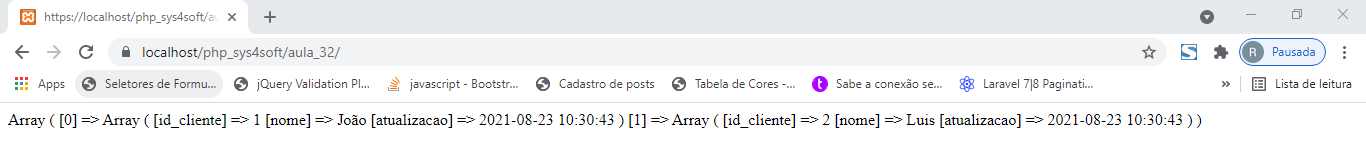
$gestor = new gestor();

$query = "select \* from clientes";

$dados = $gestor->EXE\_QUERY($query);

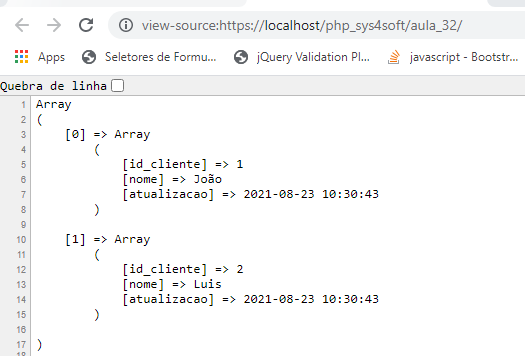
print\_r($dados);

?>



## Melhorando a visualização dos dados

- Pressione <Ctrl><U>:



view-source: é semelhante a ver o código fonte da página

**aula\_32\index2.php**

<?php

require 'config.php';

require 'gestor.php';

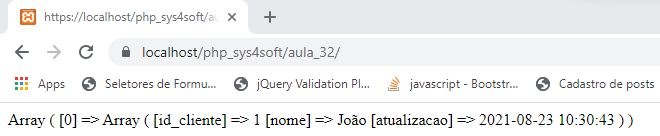
$gestor = new gestor();

$query = "select \* from clientes where id\_cliente < 2";

$dados = $gestor->EXE\_QUERY($query);

print\_r($dados);

?>



**aula\_32\index3.php**

<?php

require 'config.php';

require 'gestor.php';

$gestor = new gestor();

$query = "select \* from clientes where id\_cliente > :id";

$parametros = [

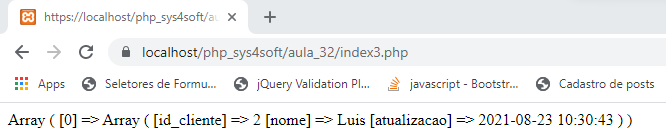
":id" => 1

];

$dados = $gestor->EXE\_QUERY($query, $parametros);

print\_r($dados);

?>



# Aula 33 - Encriptação de passwords em PHP

**aula\_33\index.php**

<?php

$pass1 = "123456";

$enc1 = md5($pass1);

echo "<p>$pass1</p><p>$enc1</p>";

echo "<p>==========================================================</p>";

$pass2 = "abcdef";

$enc2 = md5($pass2);

echo "<p>$pass2</p><p>$enc2</p>";

echo "<p>==========================================================</p>";

$pass3 = "abcdef";

$enc3 = password\_hash($pass3, PASSWORD\_DEFAULT);

echo "<p>$pass3</p><p>$enc3</p>";

echo "<p>==========================================================</p>";

$pass4 = "abcdef";

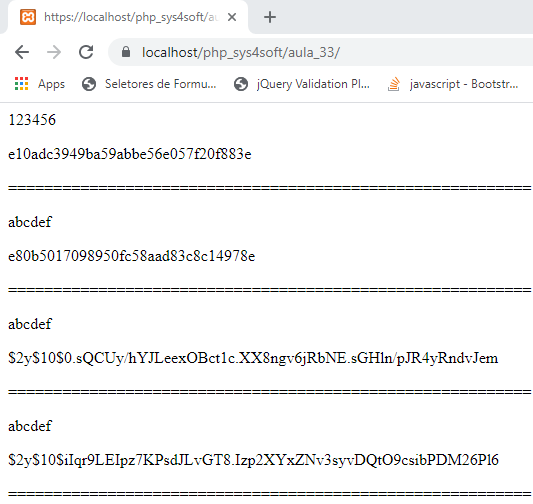
$enc4 = password\_hash($pass4, PASSWORD\_BCRYPT);

echo "<p>$pass4</p><p>$enc4</p>";

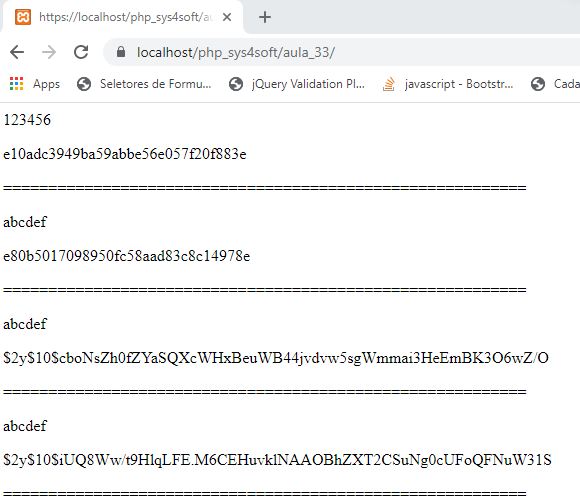
echo "<p>==========================================================</p>";

?>

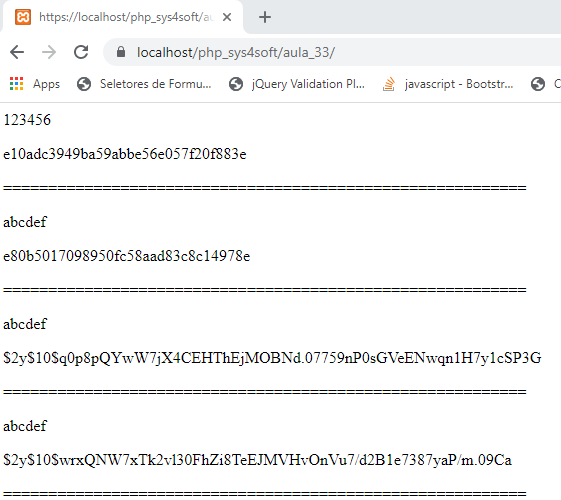
- No browser:



- Atualizando a página:



- Atualizando novamente a página:



# Aula 34 - Mecanismos de funcionamento geral de um sistema de login

## Sistema utilizando encriptação com md5()

**aula\_34\index1.php**

<?php

// no BD

$usuario = 'joao';

$pass = md5("123456");

// no formulário

$f\_usuario = 'joao';

$f\_pass = "123456";

// logica

$hash = md5($f\_pass);

// analisar o login

if($f\_usuario == $usuario && $hash == $pass){

echo "<p>Sucesso!</p>";

} else {

echo "<p>Insucesso!</p>";

}

?>



## Sistema utilizando encriptação com password\_hash()

**aula\_34\index2.php**

<?php

// no BD

$usuario = 'joao';

$pass = password\_hash("123456", PASSWORD\_DEFAULT);

// no formulário

$f\_usuario = 'joao';

$f\_pass = "123456";

// analisar o login

if($f\_usuario == $usuario && password\_verify($f\_pass, $pass)){

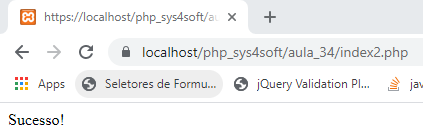
echo "<p>Sucesso!</p>";

} else {

echo "<p>Insucesso!</p>";

}

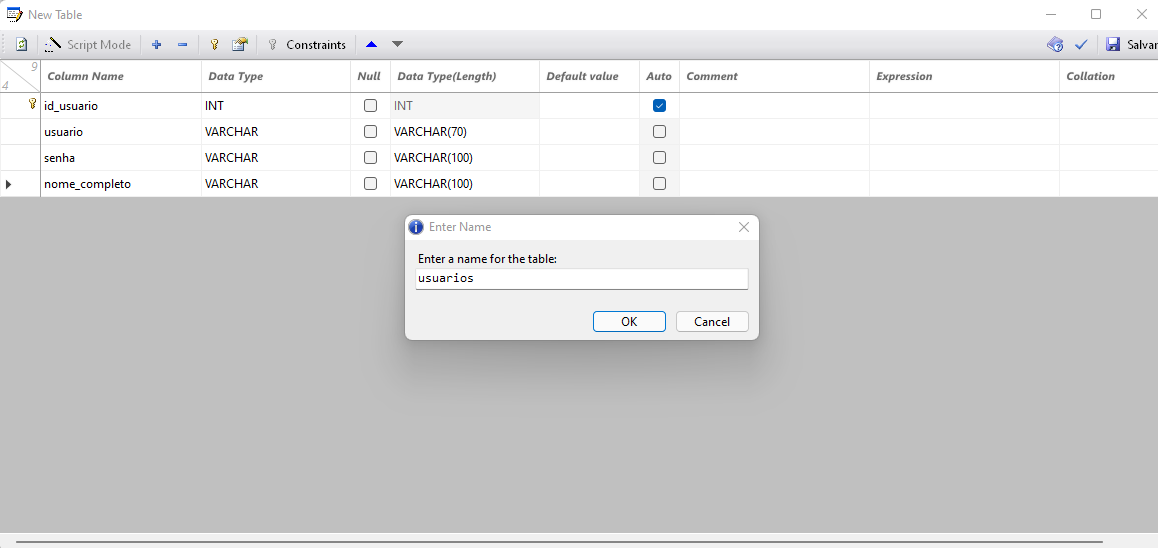
?>



# Aula 35 - Exemplo prático - Sistema de login em PHP

## Criando a base de dados

- Crie uma BD chamada clientes com uma tabela chamada usuários:



## Usando sistema md5 de encriptação

**aula\_35\config.php**

<?php

$db\_host = 'localhost';

$db\_name = 'clientes';

$db\_username = 'root';

$db\_password = '';

$db\_charset = 'utf8';

?>

**aula\_35\gestor.php**

<?php

// classe para a gestão da BD

class gestor {

// ===============================================

public function EXE\_QUERY($query, $parametros = null){

// resultados

$resultados = null;

// SELECT

// abre a conexão à BD

$conn = new PDO('mysql:host='.$GLOBALS['db\_host'].

';dbname='.$GLOBALS['db\_name'].

';charset='.$GLOBALS['db\_charset'],

$GLOBALS['db\_username'],

$GLOBALS['db\_password'],

array(PDO::ATTR\_PERSISTENT=>TRUE)

);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

// executa a query

if($parametros != null){

$gest = $conn->prepare($query);

$gest->execute($parametros);

$resultados = $gest->fetchAll(PDO::FETCH\_ASSOC); // retorna os dados através de um array associativo

} else {

$gest = $conn->prepare($query);

$gest->execute();

$resultados = $gest->fetchAll(PDO::FETCH\_ASSOC); // retorna os dados através de um array associativo

}

// fechar a conexão

$conn = null;

return $resultados;

}

// ===============================================

public function EXE\_NON\_QUERY($query, $parametros = null){

// INSERT, UPDATE, DELETE

// abre a conexão à BD

$conn = new PDO('mysql:host='.$GLOBALS['db\_host'].

';dbname='.$GLOBALS['db\_name'].

';charset='.$GLOBALS['db\_charset'],

$GLOBALS['db\_username'],

$GLOBALS['db\_password'],

array(PDO::ATTR\_PERSISTENT=>TRUE)

);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

// executa a query

$conn->beginTransaction();

try{

if($parametros != null){

$gest = $conn->prepare($query);

$gest->execute($parametros);

} else {

$gest = $conn->prepare($query);

$gest->execute();

}

$conn->commit();

} catch(PDOException $e){

echo '<p>' . $e . '</p>';

$conn->rollBack();

}

$conn = null;

}

}

?>

**aula\_35\cadastrar\_usuario.php**

<?php

$usuario = 'admin';

$senha = 'abc';

$nome\_completo = 'Administrador';

require 'config.php';

require 'gestor.php';

// parâmetros do usuário

$parametros = [

":usuario" => $usuario,

":senha" => md5($senha),

":nome\_completo" => $nome\_completo

];

$query = "insert into usuarios (usuario, senha, nome\_completo) values (:usuario, :senha, :nome\_completo)";

// inserir usuário na BD

$gest = new gestor();

$gest->EXE\_NON\_QUERY($query, $parametros);

echo "<p>Usuário cadastrado com sucesso!</p>";

?>

**aula\_35\formulario.php**

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

<label>Usuário:</label><br />

<input type="text" name="txt\_usuario" /><br />

<label>Senha:</label><br />

<input type="password" name="txt\_senha" /><br /><br />

<input type="submit" value="Entrar" />

</form>

**aula\_35\verificar.php**

<?php

// verifica se os dados estão corretos

$usuario = $\_POST['txt\_usuario'];

$senha = $\_POST['txt\_senha'];

require 'config.php';

require 'gestor.php';

$parametros = [

":usuario" => $usuario,

":senha" => md5($senha)

];

$query = "select \* from usuarios where usuario = :usuario and senha = :senha";

$gest = new gestor();

$resultados = $gest->EXE\_QUERY($query, $parametros);

if(count($resultados) == 0){

echo '<p>Login inválido!</p>';

} else {

echo '<p>Login válido!</p>';

// entrar na plataforma

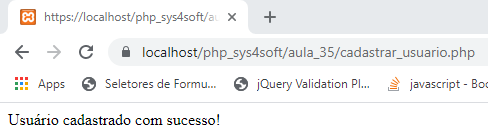
}

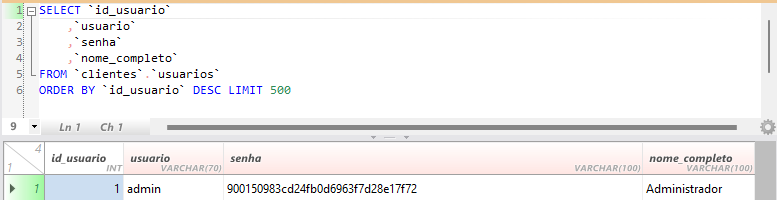
?>

## Cadastrando um usuário

- No browser:

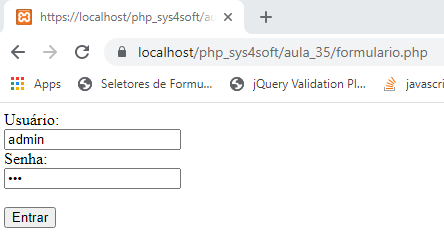
localhost/ php\_sys4soft/aula35/cadastrar\_usuario.php

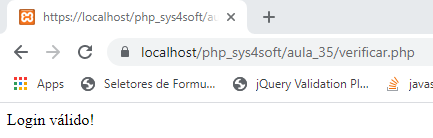




## Fazendo o login

https://localhost/php\_sys4soft/aula\_35/formulario.php





# Aula 36 - Sessões em PHP - Parte1

- Cadastre 2 usuários:

**aula\_36\cadastrar\_usuario.php**

<?php

$usuario = 'joao';

$senha = 'aaaa';

$nome\_completo = 'João Ribeiro';

require 'config.php';

require 'gestor.php';

// parâmetros do usuário

$parametros = [

":usuario" => $usuario,

":senha" => md5($senha),

":nome\_completo" => $nome\_completo

];

$query = "insert into usuarios (usuario, senha, nome\_completo) values (:usuario, :senha, :nome\_completo)";

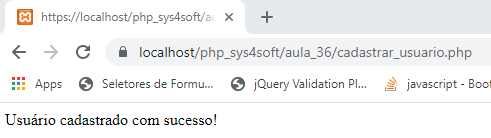
// inserir usuário na BD

$gest = new gestor();

$gest->EXE\_NON\_QUERY($query, $parametros);

echo "<p>Usuário cadastrado com sucesso!</p>";

?>



**aula\_36\cadastrar\_usuario.php**

<?php

$usuario = 'luis';

$senha = 'bbbb';

$nome\_completo = 'Luis Dias';

require 'config.php';

require 'gestor.php';

// parâmetros do usuário

$parametros = [

":usuario" => $usuario,

":senha" => md5($senha),

":nome\_completo" => $nome\_completo

];

$query = "insert into usuarios (usuario, senha, nome\_completo) values (:usuario, :senha, :nome\_completo)";

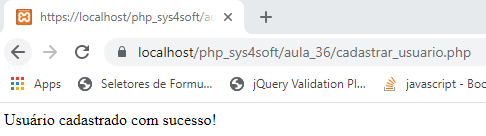
// inserir usuário na BD

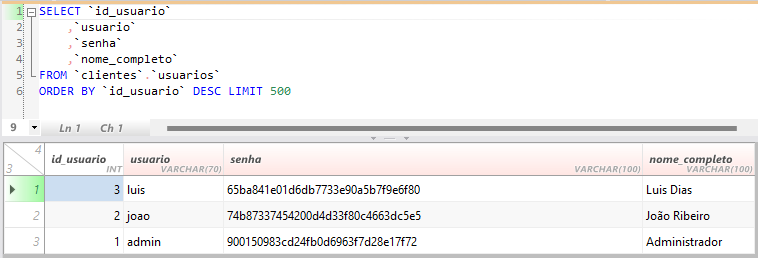
$gest = new gestor();

$gest->EXE\_NON\_QUERY($query, $parametros);

echo "<p>Usuário cadastrado com sucesso!</p>";

?>





**aula\_36\index.php**

<?php

// abertura da plataforma

session\_start();

if(isset($\_SESSION['nome'])){

echo '<p>' . $\_SESSION['nome'] . '</p>';

}

?>

<h3>LOGIN</h3>

<?php

// inserir o formulário

require 'formulario.php';

?>

**aula\_36\verificar.php**

<?php

session\_start();

// verifica se os dados estão corretos

$usuario = $\_POST['txt\_usuario'];

$senha = $\_POST['txt\_senha'];

require 'config.php';

require 'gestor.php';

$parametros = [

":usuario" => $usuario,

":senha" => md5($senha)

];

$query = "select \* from usuarios where usuario = :usuario and senha = :senha";

$gest = new gestor();

$resultados = $gest->EXE\_QUERY($query, $parametros);

if(count($resultados) == 0){

echo '<p>Login inválido!</p><p><a href="index.php">Voltar</a></p>';

} else {

$\_SESSION['nome'] = $resultados[0]['nome\_completo'];

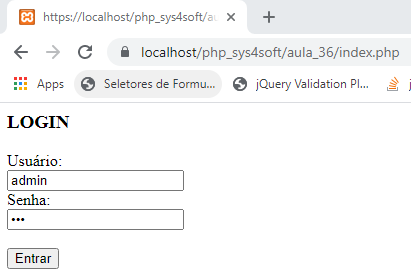
echo '<p>Login válido!</p><p><a href="index.php">Voltar</a></p>';

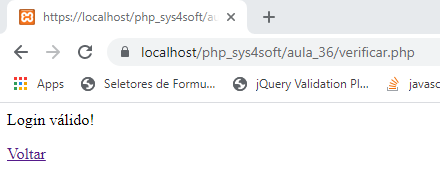
}

?>

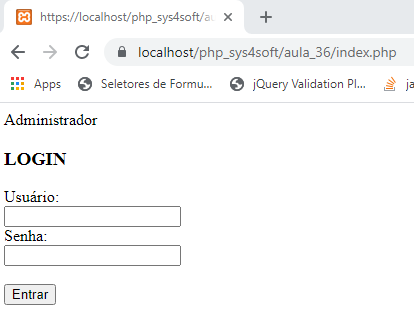
## Fazendo o login

https://localhost/php\_sys4soft/aula\_36/index.php





Ao clicar no botão "Voltar":



# Aula 37 - Sessões em PHP - Parte2

**aula\_37\formulario.php**

<h3>LOGIN</h3>

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

<label>Usuário:</label><br />

<input type="text" name="txt\_usuario" /><br />

<label>Senha:</label><br />

<input type="password" name="txt\_senha" /><br /><br />

<input type="submit" value="Entrar" />

</form>

**aula\_37\index.php**

<?php

// abertura da plataforma

ini\_set('session.cookie.httponly', true);

session\_start();

if(isset($\_SESSION['last\_IP']) === false){

$\_SESSION['last\_IP'] = $\_SERVER['REMOTE\_ADDR'];

}

if($\_SESSION['last\_IP'] !== $\_SERVER['REMOTE\_ADDR']){

session\_destroy();

}

$sessao = false;

if(isset($\_SESSION['nome'])){

$sessao = true;

}

if(!$sessao){

// inserir o formulário

require 'formulario.php';

} else {

echo '<p>Bem vindo ' . $\_SESSION['nome'] . ' | <a href="logout.php">Logout</a></p>';

}

?>

**aula\_37\logout.php**

<?php

// logout

ini\_set('session.cookie.httponly', true);

session\_start();

if(isset($\_SESSION['last\_IP']) === false){

$\_SESSION['last\_IP'] = $\_SERVER['REMOTE\_ADDR'];

}

if($\_SESSION['last\_IP'] !== $\_SERVER['REMOTE\_ADDR']){

session\_destroy();

}

echo '<p>Até a próxima, ' . $\_SESSION['nome'] . '!</p>';

session\_unset();

session\_destroy();

echo '<p><a href="index.php">Voltar</a></p>';

?>

**aula\_37\verificar.php**

<?php

ini\_set('session.cookie.httponly', true);

session\_start();

if(isset($\_SESSION['last\_IP']) === false){

$\_SESSION['last\_IP'] = $\_SERVER['REMOTE\_ADDR'];

}

if($\_SESSION['last\_IP'] !== $\_SERVER['REMOTE\_ADDR']){

session\_destroy();

}

// verifica se os dados estão corretos

$usuario = $\_POST['txt\_usuario'];

$senha = $\_POST['txt\_senha'];

require 'config.php';

require 'gestor.php';

$parametros = [

":usuario" => $usuario,

":senha" => md5($senha)

];

$query = "select \* from usuarios where usuario = :usuario and senha = :senha";

$gest = new gestor();

$resultados = $gest->EXE\_QUERY($query, $parametros);

if(count($resultados) == 0){

echo '<p>Login inválido!</p><p><a href="index.php">Voltar</a></p>';

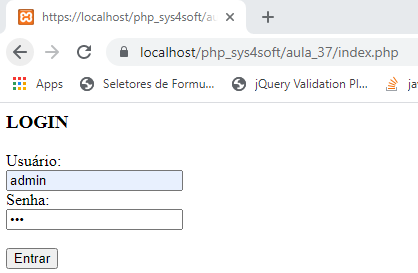
} else {

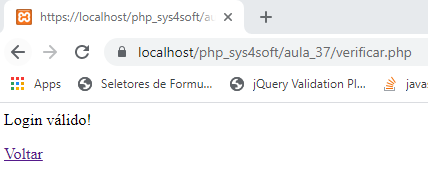
$\_SESSION['nome'] = $resultados[0]['nome\_completo'];

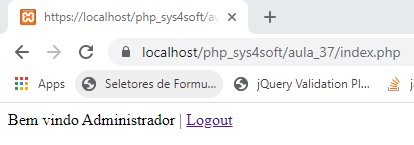
echo '<p>Login válido!</p><p><a href="index.php">Voltar</a></p>';

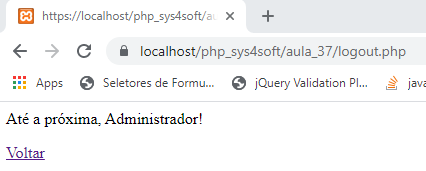
}

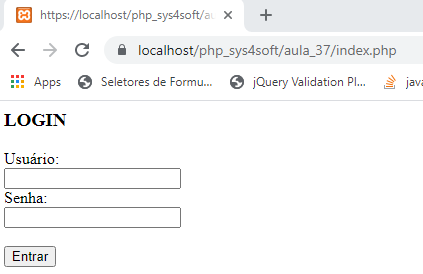
?>



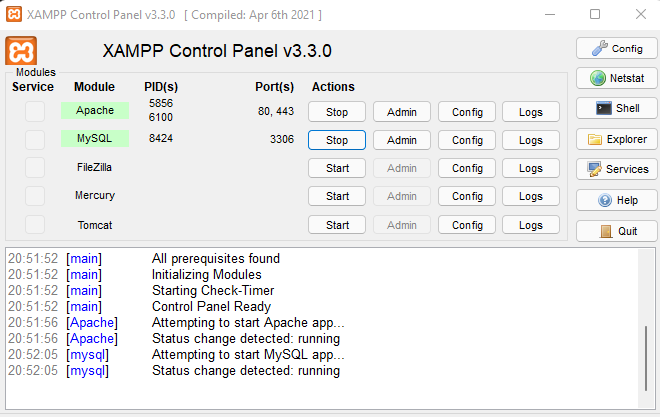








# Aula 38 - Permissões de acesso a base de dados



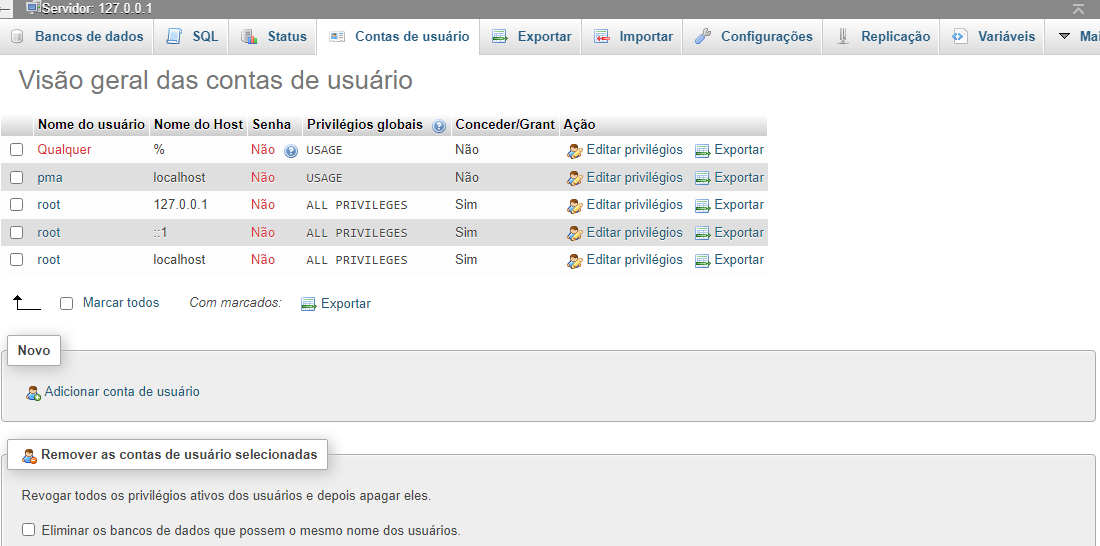
## Database .NET

- No BD mysql procure pela tabela user. Nesta tabela estão registradas as permissões dos usuários.

## Phpmyadmin

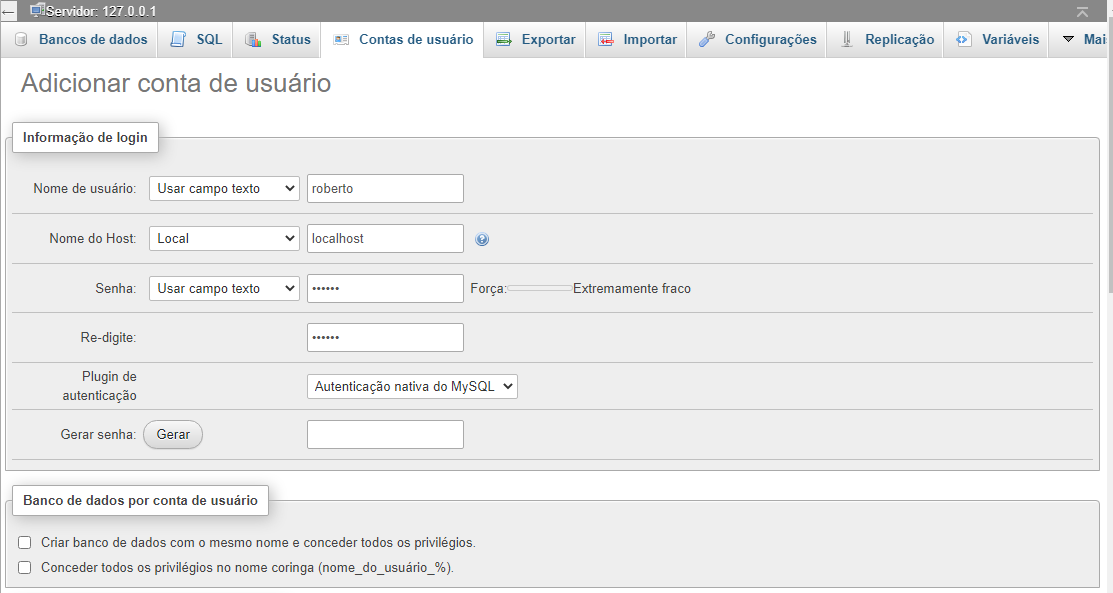
- Abra o phpmyadmin.

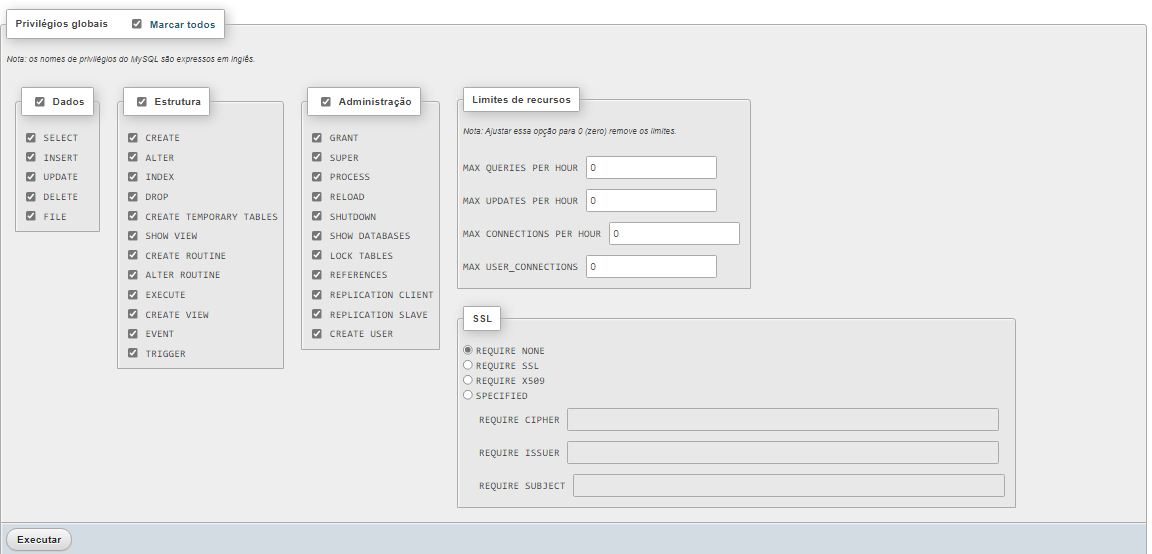
- Clique na aba "Contas de usuário"



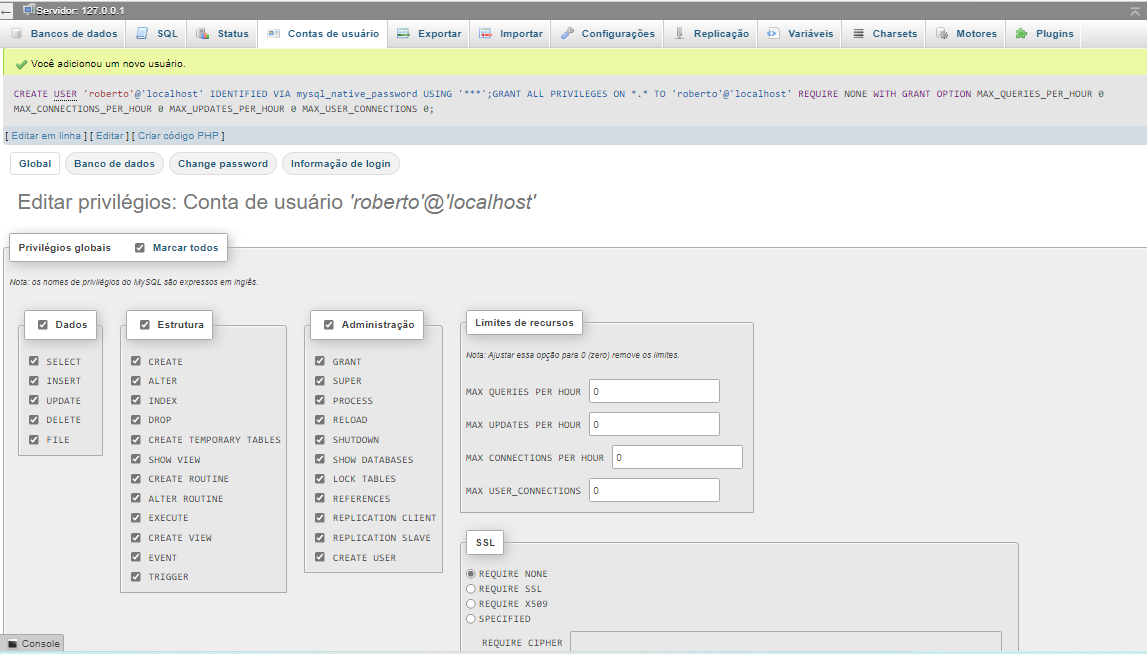
- Clique em "Adicionar conta de usuário"

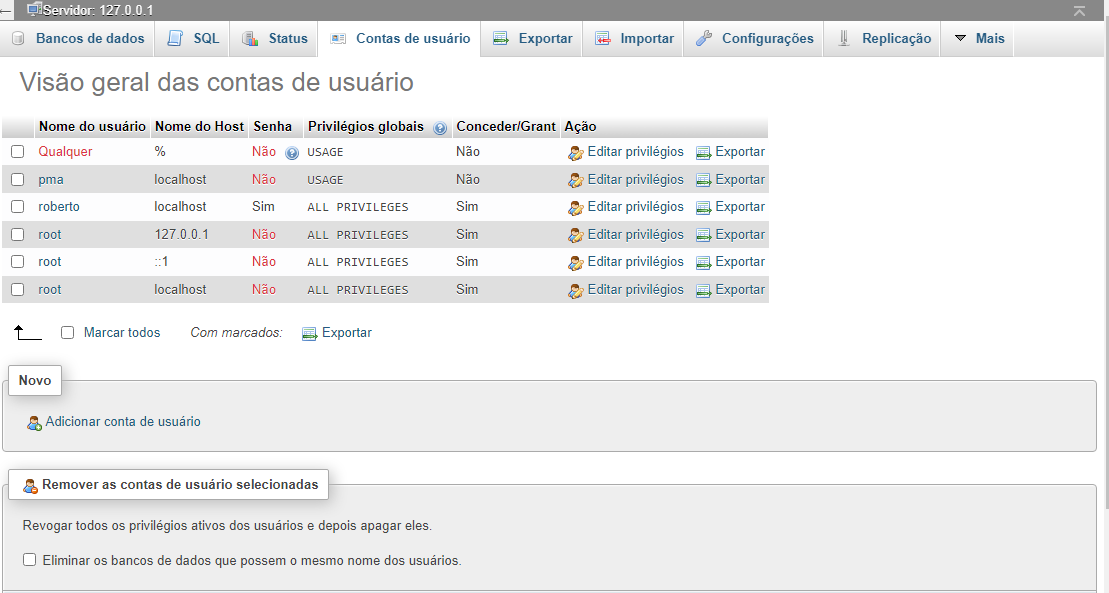
Senha: 123456





- Clique no botão "Executar"





- Alterar as linhas seguintes do arquivo config.inc.php

**config.inc.php**

<?php

/\*

\* This is needed for cookie based authentication to encrypt password in

\* cookie

\*/

$cfg['blowfish\_secret'] = 'xampp'; /\* YOU SHOULD CHANGE THIS FOR A MORE SECURE COOKIE AUTH! \*/

/\*

\* Servers configuration

\*/

$i = 0;

/\*

\* First server

\*/

$i++;

/\* Authentication type and info \*/

/\* $cfg['Servers'][$i]['auth\_type'] = 'config'; \*/

$cfg['Servers'][$i]['auth\_type'] = 'cookie';

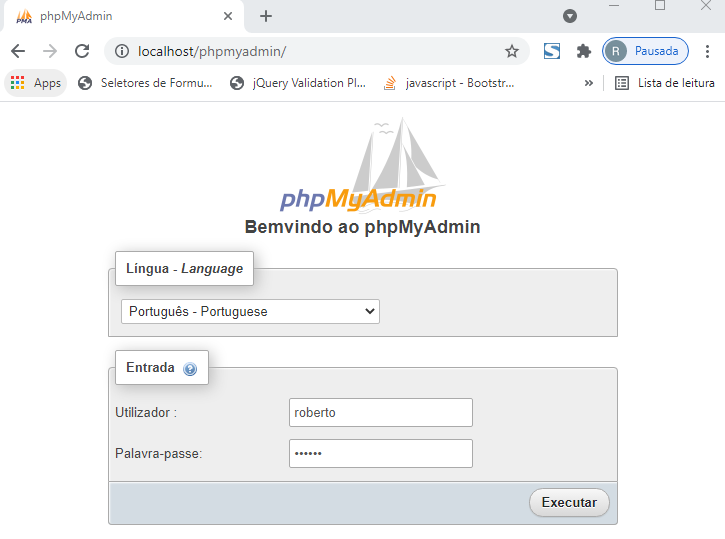
$cfg['Servers'][$i]['user'] = 'root';

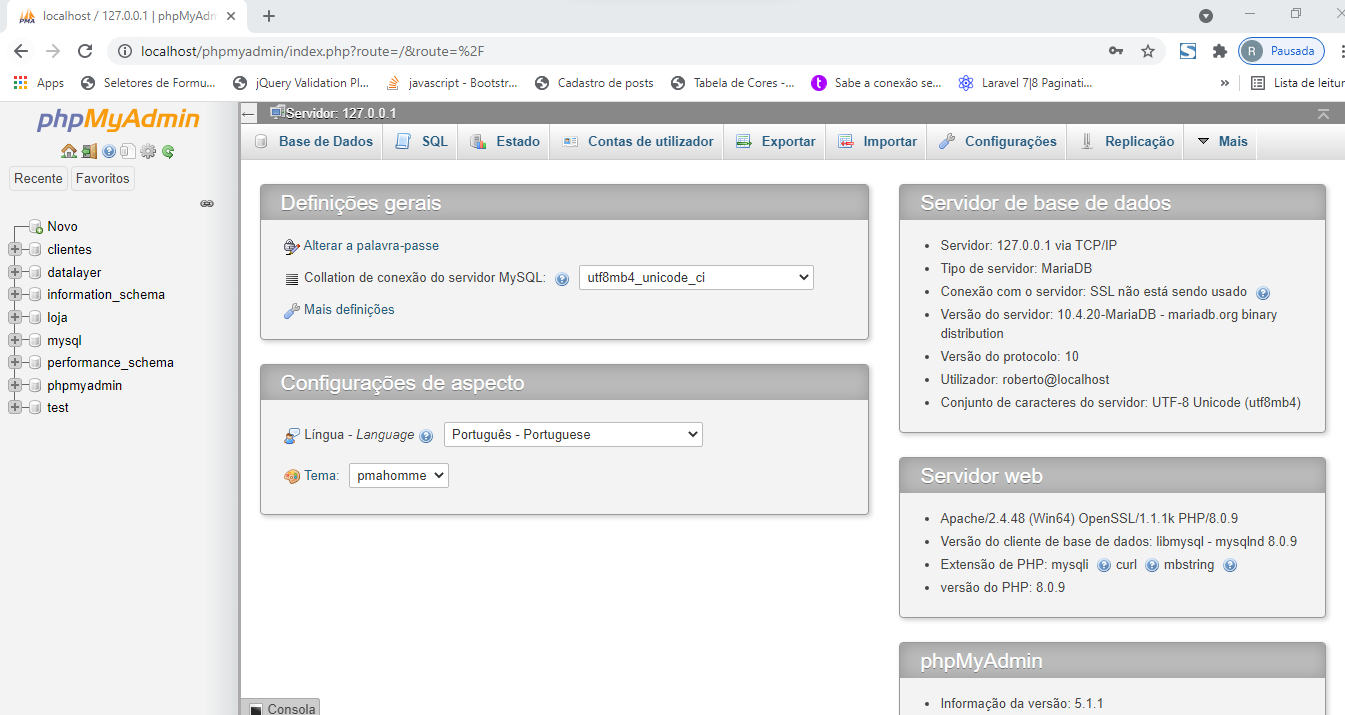
$cfg['Servers'][$i]['password'] = '';

$cfg['Servers'][$i]['extension'] = 'mysqli';

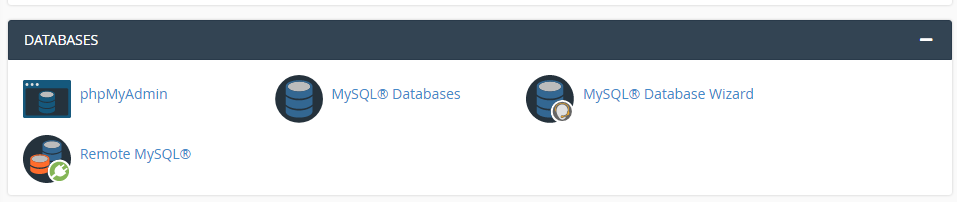
$cfg['Servers'][$i]['AllowNoPassword'] = true;

$cfg['Lang'] = '';

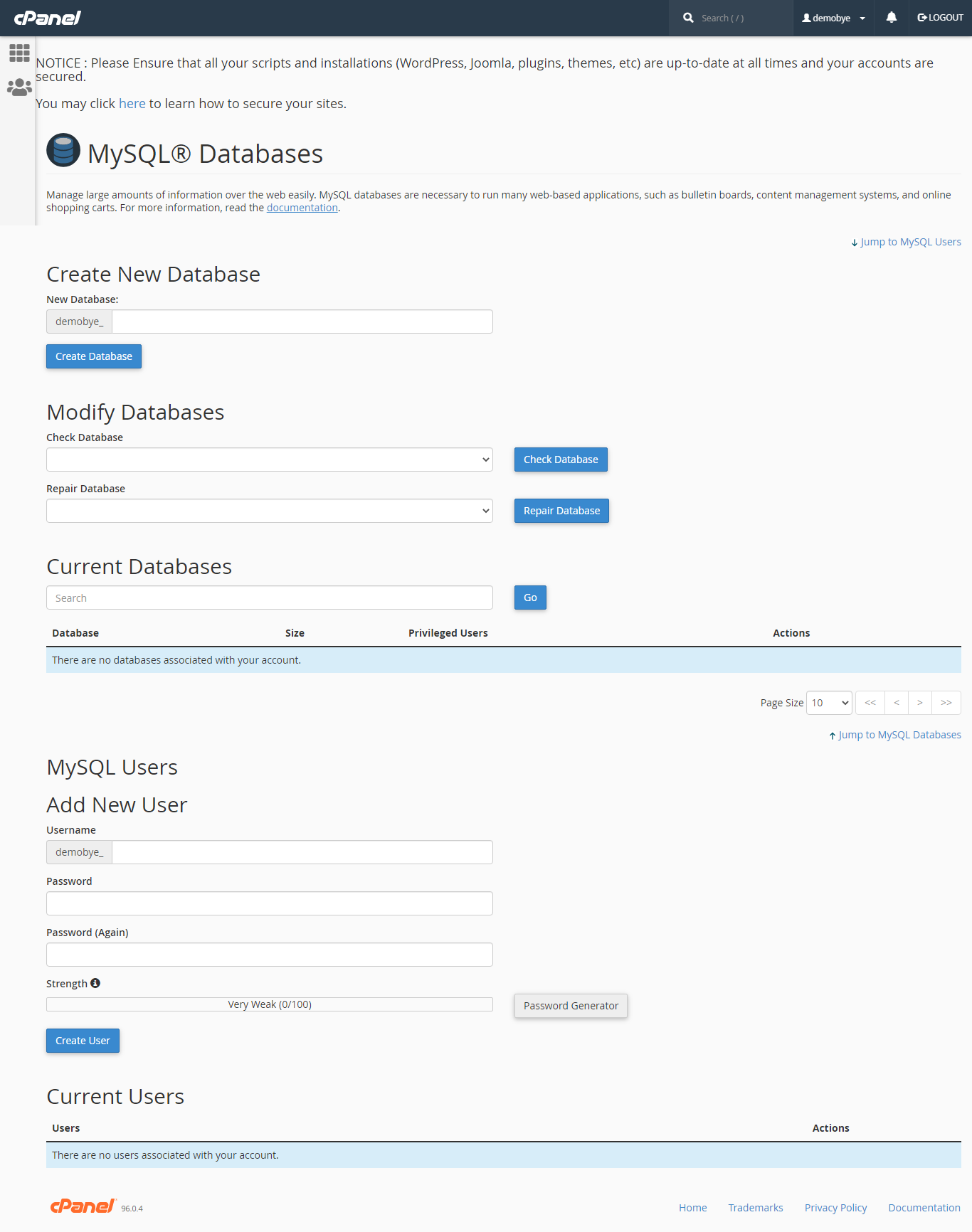




# Aula 39 - Criação de usuário mysql em cpanel



- Clique em MySQL & Databases:



# Aula 40 - Como enviar emails em PHP - parte1

- Baixe a versão PHPMailer-5.2.21

- Insira seus arquivos na pasta C:\xampp\htdocs\php\_sys4soft\aula\_40

- Clique na pasta examples e abra o arquivo gmail.phps

- Apague os comentários

- Crie uma função chamada EnviarEmail e coloque o conteudo deste arquivo dentro dela:

**aula\_40\examples\gmail.phps**

<?php

function EnviarEmail(){

date\_default\_timezone\_set('Etc/UTC');

require '../PHPMailerAutoload.php';

$mail = new PHPMailer;

$mail->isSMTP();

$mail->SMTPDebug = 2;

$mail->Debugoutput = 'html';

$mail->Host = 'smtp.gmail.com';

$mail->Port = 587;

$mail->SMTPSecure = 'tls';

$mail->SMTPAuth = true;

$mail->Username = "username@gmail.com";

$mail->Password = "yourpassword";

$mail->setFrom('from@example.com', 'First Last');

$mail->addReplyTo('replyto@example.com', 'First Last');

$mail->addAddress('whoto@example.com', 'John Doe');

$mail->Subject = 'PHPMailer GMail SMTP test';

$mail->msgHTML(file\_get\_contents('contents.html'), dirname(\_\_FILE\_\_));

$mail->AltBody = 'This is a plain-text message body';

$mail->addAttachment('images/phpmailer\_mini.png');

//send the message, check for errors

if (!$mail->send()) {

echo "Mailer Error: " . $mail->ErrorInfo;

} else {

echo "Message sent!";

}

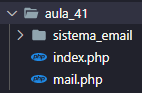
}

# Aula 41 - Como enviar emails em PHP - parte2

- Baixe a versão PHPMailer-5.2.21

- Copie a pasta PHPMailer-5.2.21 para C:\xampp\htdocs\php\_sys4soft\aula\_41

- Renomeie essa pasta para sistema\_email



**aula\_41\mail.php**

<?php

function EnviarEmail($to, $from, $assunto, $mensagem){

// envio de email através de conta do GMail

require 'sistema\_email/PHPMailerAutoload.php';

$mail = new PHPMailer;

$mail->CharSet = "UTF-8";

$mail->Host = 'smtp.gmail.com';

$mail->Port = 587;

$mail->isSMTP();

$mail->SMTPSecure = 'tls';

$mail->SMTPAuth = true;

// sistema de informação do erro

$mail->SMTPDebug = 0;

$mail->isHTML(true);

$mail->Username = "robertopinheiro7843@gmail.com"; // insira o seu email

$mail->Password = "\*\*\*\*\*\*\*\*\*\*"; // insira a sua senha

$mail->setFrom($from, $from);

$mail->addAddress($to, $to);

// assunto e mensagem de email

$mail->Subject = $assunto;

$mail->Body = $mensagem;

//send the message, check for errors

if (!$mail->send()) {

echo "Mailer Error: " . $mail->ErrorInfo;

} else {

echo "Message sent!";

}

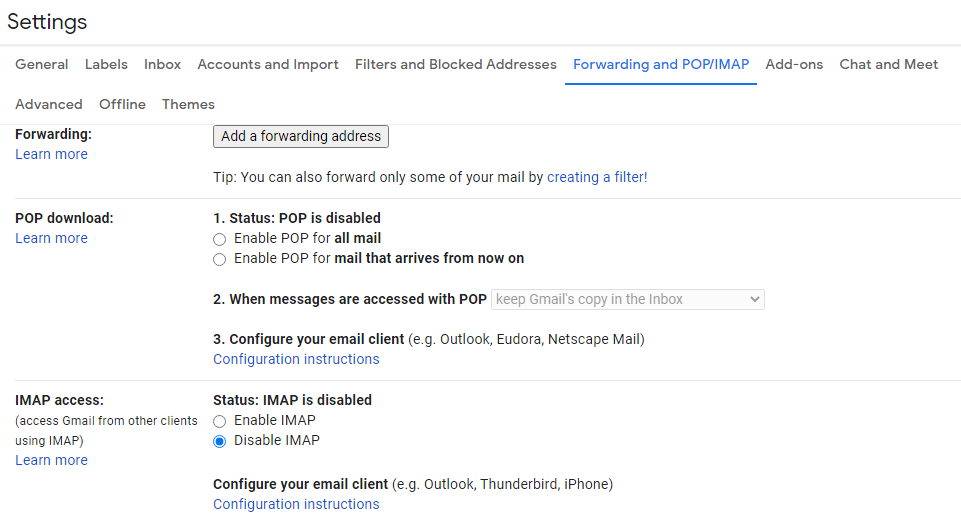
}

?>

- Acesse sua conta no gmail.

- Em Settings:

- Clique na aba: Forwarding and POP/IMAP



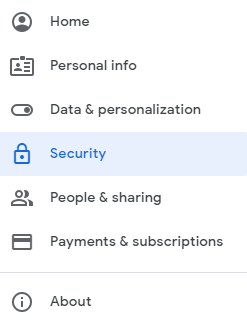
- Clique no link Configuration instructions



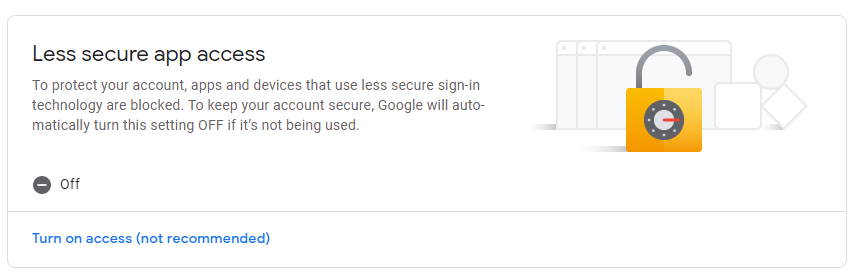
- Clique na aba Accounts and Import:

- Clique em Other Google Account settings

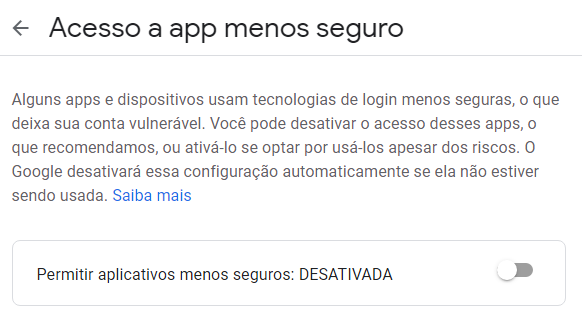
- Clique em Security



Allow less secure apps: OFF

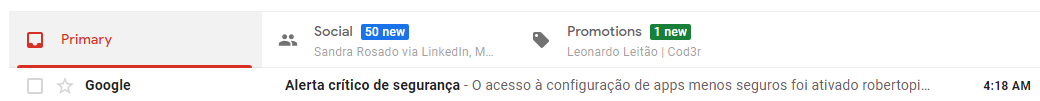


- Clique no link Turn on access (not recommended)



- Para funcionar, obrigatoriamente ative essa opção:





## Enviando o email

- Crie um arquivo chamado index.php

**aula\_41\index.php**

<?php

require 'mail.php';

$mensagem = "<p>Este é um parágrafo de HTML <strong> - MUITO BEM - </strong></p><br />Esta é uma nova linha.";

EnviarEmail("robertopinheiro7843@gmail.com", "robertopinheiro7843@gmail.com", "Mensagem de teste", $mensagem);

?>

- Comente as seguintes linhas do arquivo aula\_41\sistema\_email\PHPMailerAutoload.php:

**aula\_41\sistema\_email\PHPMailerAutoload.php**

<?php

/\*\*

\* PHPMailer SPL autoloader.

\* PHP Version 5

\* @package PHPMailer

\* @link https://github.com/PHPMailer/PHPMailer/ The PHPMailer GitHub project

\* @author Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>

\* @author Jim Jagielski (jimjag) <jimjag@gmail.com>

\* @author Andy Prevost (codeworxtech) <codeworxtech@users.sourceforge.net>

\* @author Brent R. Matzelle (original founder)

\* @copyright 2012 - 2014 Marcus Bointon

\* @copyright 2010 - 2012 Jim Jagielski

\* @copyright 2004 - 2009 Andy Prevost

\* @license http://www.gnu.org/copyleft/lesser.html GNU Lesser General Public License

\* @note This program is distributed in the hope that it will be useful - WITHOUT

\* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or

\* FITNESS FOR A PARTICULAR PURPOSE.

\*/

/\*\*

\* PHPMailer SPL autoloader.

\* @param string $classname The name of the class to load

\*/

function PHPMailerAutoload($classname)

{

//Can't use \_\_DIR\_\_ as it's only in PHP 5.3+

$filename = dirname(\_\_FILE\_\_).DIRECTORY\_SEPARATOR.'class.'.strtolower($classname).'.php';

if (is\_readable($filename)) {

require $filename;

}

}

if (version\_compare(PHP\_VERSION, '5.1.2', '>=')) {

//SPL autoloading was introduced in PHP 5.1.2

if (version\_compare(PHP\_VERSION, '5.3.0', '>=')) {

spl\_autoload\_register('PHPMailerAutoload', true, true);

} else {

spl\_autoload\_register('PHPMailerAutoload');

}

} else {

/\*\*

\* Fall back to traditional autoload for old PHP versions

\* @param string $classname The name of the class to load

\*/

// function \_\_autoload($classname)

// {

// PHPMailerAutoload($classname);

// }

}

- No browser:

http://localhost/php\_sys4soft/aula\_41/

