



Workshop PHP

Nível Básico

18/06/2016

Agenda

09:15 - 09:20	Sobre vocês
09:20 - 09:22	Sobre nós
09:22 - 09:25	Introdução ao protocolo HTTP
09:25 - 09:45	Como o HTTP funciona em relação as aplicações web
09:45 - 10:00	Métodos do HTTP para web (GET, POST...)
10:00 - 10:40	Formas de realizar as requisições (Formulários e Ajax)
10:40 - 11:20	Conexão com MySQL
11:20 - 12:00	Criação de um sistema de cadastro e <i>login</i>
12:00 - 13:00	Dúvidas

Sobre vocês

Sobre vocês

Contem um pouco sobre vocês!

1. Nome
2. O que faz da vida
3. Experiência com Web
4. Experiência com PHP
5. O que espera desse Workshop

Sobre nós

Sobre nós / Colaboradores



Guilherme Uezima

Graduado em Sistemas de Informação pelo Mackenzie, atua há mais de cinco anos com o desenvolvimento de soluções tecnológicas para diversas plataformas.



Gustavo Soré

Graduado em Sistemas de Informação no Mackenzie. Desenvolvedor de aplicativos, especialista em iOS. Responsável por aplicativos de CRM e mídia digital para o setor farmacêutico na América Latina.



Michel Zarzour F.

Graduado em Sistemas de Informação pelo Mackenzie, possui grande experiência na área de desenvolvimento de aplicações WebApp e participação em grupos de startup.



Otávio R. Rossi

Graduado em Sistemas de Informação pelo Mackenzie, possui grande experiência na área de desenvolvimento de aplicações Web e consultoria de negócios na área de tecnologia em grandes empresas do mercado

Sobre nós



Tecnologias e ferramentas modernas para desenvolvimento de soluções em cloud



Gerenciamento dos feedbacks, recomendações de mercado e implementação de melhorias



Foco na experiência do usuário na utilização das soluções customizadas em multiplataformas



<http://inopus.com.br>

Introdução ao protocolo HTTP

Introdução ao protocolo HTTP

Hypertext Transfer Protocol

“Is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World-Wide Web global information initiative since 1990. The first version of HTTP, referred to as HTTP/0.9, was a simple protocol for raw data transfer across the Internet”.

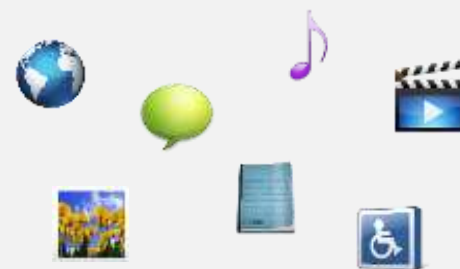
IETF



- Camada de aplicação
- *Hypermedia*
- Usado pela WWW (*World-Wide Web*)
- Atualmente da versão 2.0 (HTTP/2)

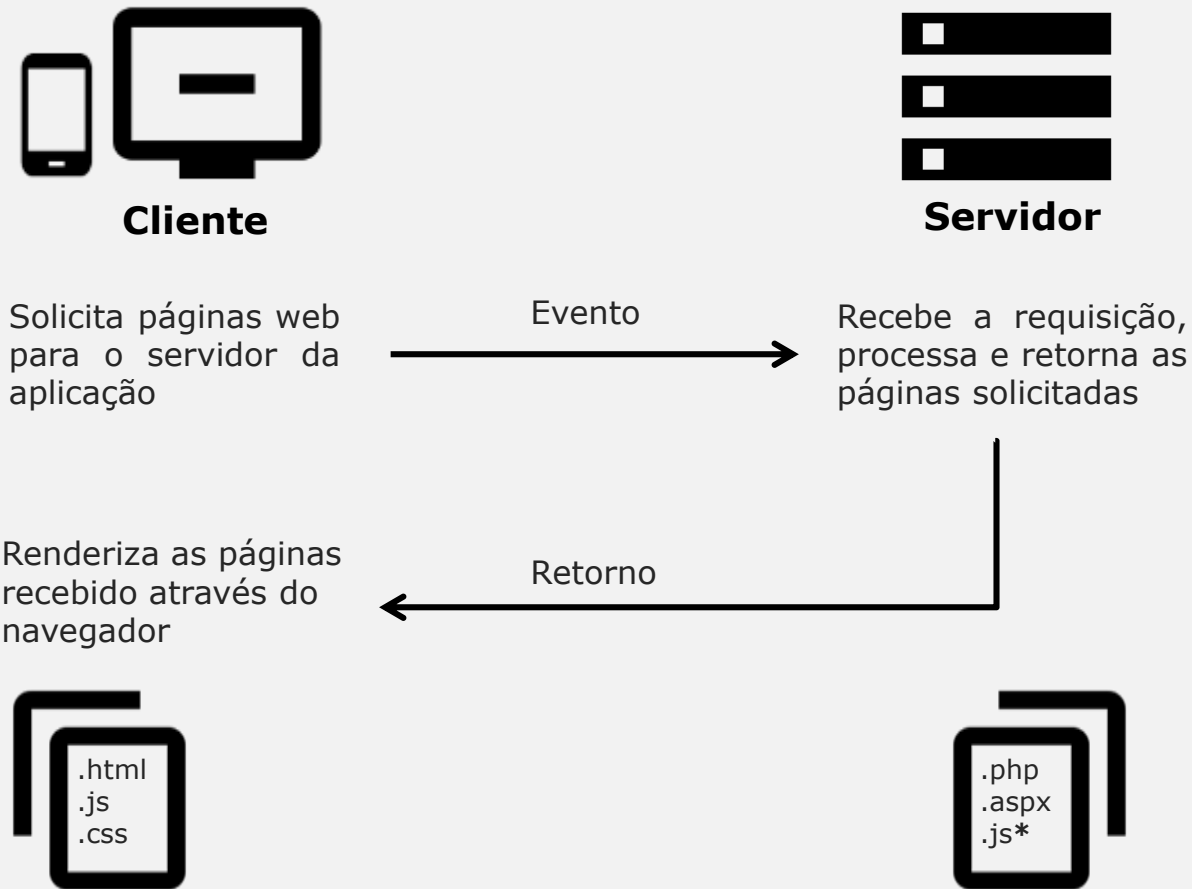


Utilização:



Como o HTTP funciona em relação as aplicações web

Como o HTTP funciona em relação as aplicações web



Hello World!

hello_world.html

```
01 <h1>hello world!</h1>
```

```
02
```

```
03
```

```
04
```

```
05
```

```
06
```

```
07
```

```
08
```

```
09
```

```
10
```

```
11
```

```
12
```

```
13
```

```
14
```

```
15
```

```
16
```

```
17
```

Hello World!

hello_world.php

```
01 <h1>
02 <?php
03     echo "Hello World!!";
04     echo ("Hello World!! - Assim também funciona!!");
05     print ("E que tal assim?");
06 ?>
07 </h1>
08
09
10
11
12
13
14
15
16
17
```

Métodos do HTTP para web (GET, POST....)

Métodos do HTTP para web (GET, POST...)

O HTTP possui alguns métodos que indicam a ação requerida com aquela requisição. Dentre esses diferentes tipos de métodos existentes, existem 5 que são os mais utilizados e difundidos.

<i>Method</i>	<i>Request Body</i>	<i>Response Body</i>
GET	No	Yes
HEAD	No	No
POST	Yes	Yes
PUT	Yes	Yes
DELETE	No	Yes

Métodos do HTTP para web (GET, POST...)

1

GET <https://www.google.com.br/search?q=tecnodrom&ie=utf-8&oe=utf-8&client=firefox-b-ab>

2

GET <http://tecnodrom.com/Postagens/380/inopus-vence-hackathon-da-ibm>

3

POST <https://rapordo.com/usuario/entrar>

usuario: otavio
senha: af12354abcd134fed

Formas de realizar as requisições (Formulários e Ajax)

Formas de realizar as requisições (Formulários e Ajax)

Formulários

1. `<form>` no HTML representa uma seção do documento que contém elementos interativos para serem enviados ao servidor;
2. Principais atributos do `<form>`:
 - 2.1. `action` – URL que receberá o formulário;
 - 2.2. `enctype` – utilizado no POST, *MIME Type* do conteúdo;
 - 2.3. `method` – Método HTTP que será utilizado (GET ou POST).

Ajax

1. Asynchronous JavaScript + XML;
2. Termo utilizado para descrever uma “nova” (2005) abordagem na utilização de diversas tecnologias;
3. XMLHttpRequest();
 - 3.1. API para transferência de dados;
 - 3.2. Modos assíncronos e síncronos;
 - 3.3. Utilizado principalmente para JSON.

Formas de realizar as requisições (Formulários e Ajax)

formulario_get.html

```
01 <form action="servico_get.php" method="get">
02     <label for="usuario">Usuário</label>
03     <br>
04     <input type="text" name="usuario" id="usuario">
05     <br>
06     <label for="nome">Nome</label>
07     <br>
08     <input type="text" name="nome" id="nome">
09     <br>
10     <input type="submit" value="Enviar">
11 </form>
```

Formas de realizar as requisições (Formulários e Ajax)

servico_get.php

```
01 <?php
02     echo $_GET["usuario"] . ' - ' . $_GET["nome"];
03 ?>
04
05
06
07
08
09
10
11
12
13
14
16
17
17
```

Formas de realizar as requisições (Formulários e Ajax)

formulario_post.html

```
01 <form action="servico_post.php" method="post">
02     <label for="usuario">Usuário</label>
03     <br>
04     <input type="text" name="usuario" id="usuario">
05     <br>
06     <label for="nome">Nome</label>
07     <br>
08     <input type="text" name="nome" id="nome">
09     <br>
10     <input type="submit" value="Enviar">
11 </form>
```

Formas de realizar as requisições (Formulários e Ajax)

servico_post.php

```
01 <?php
02     echo $_POST["usuario"] . ' - ' . $_POST["nome"];
03 ?>
04
05
06
07
08
09
10
11
12
13
14
16
17
17
```

Formas de realizar as requisições (Formulários e Ajax)

ajax_get.js

```
01 function enviaFormulario(){
02     // Obter os dados digitados pelo usuário nos campos da tela
03     var usuario_campo = document.getElementById("usuario").value;
04     var nome_campo = document.getElementById("nome").value;
05     var xhr = new XMLHttpRequest(); // Instancio um novo objeto XMLHttpRequest
06     xhr.onreadystatechange = function() {
07         // Verifico os estados da requisição (e.g. se a requisição retornou êxito)
08         if (xhr.readyState == 4 && xhr.status == 200) {
09             alert(xhr.responseText); // Caso o retorno seja sucesso
10         }
11     };
12     xhr.open("GET", "servico_get.php?usuario=" + usuario_campo + "&nome=" + nome_campo,
13     true);
14     xhr.send();
15 }
16
```

Formas de realizar as requisições (Formulários e Ajax)

ajax_get.html

```
01 <script src="ajax_get.js" type="text/javascript"></script>
02
03 <form>
04     <label for="usuario">Usuário</label>
05     <br>
06     <input type="text" name="usuario" id="usuario">
07     <br>
08     <label for="nome">Nome</label>
09     <br>
10     <input type="text" name="nome" id="nome">
11     <br>
12     <input type="button" onclick="enviaFormulario()" value="Enviar">
13 </form>
14
16
17
17
```


Formas de realizar as requisições (Formulários e Ajax)

ajax_post.js

```
01 function enviaFormulario(){
02     // Obter os dados digitados pelo usuário nos campos da tela
03     var usuario_campo = document.getElementById("usuario").value;
04     var senha_campo = document.getElementById("nome").value;
05     var data = new FormData();
06     data.append('usuario', usuario_campo);
07     data.append('nome', senha_campo);
08     var xhr = new XMLHttpRequest(); // Instancio um novo objeto XMLHttpRequest
09     xhr.onreadystatechange = function() {
10         // Verifico os estados da requisição (e.g. se a requisição retornou êxito)
11         if (xhr.readyState == 4 && xhr.status == 200) {
12             alert(xhr.responseText); // Caso o retorno seja sucesso
13         }
14     };
15     xhr.open("POST", "servico_post.php", true);
16     xhr.send(data);
17 }
```

Formas de realizar as requisições (Formulários e Ajax)

ajax_post.html

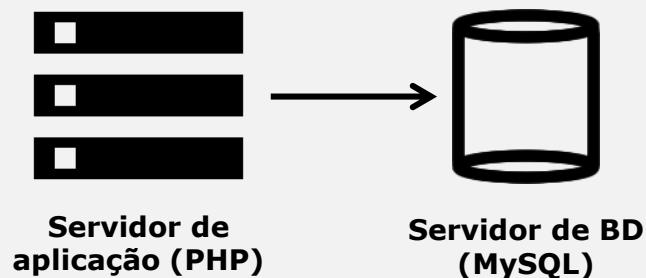
```
01 <script src="ajax_post.js" type="text/javascript"></script>
02
03 <form>
04     <label for="usuario">Usuário</label>
05     <br>
06     <input type="text" name="usuario" id="usuario">
07     <br>
08     <label for="nome">Nome</label>
09     <br>
10     <input type="text" name="nome" id="nome">
11     <br>
12     <input type="button" onclick="enviaFormulario()" value="Enviar">
13 </form>
14
16
17
17
```

Conexão com MySQL

Conexão com MySQL

MySQL

Segundo banco de dados mais popular do mundo, é um RDBMS (Relation DataBase Management System) open-source, desenvolvido, distribuído e suportado pela Oracle Corporation.



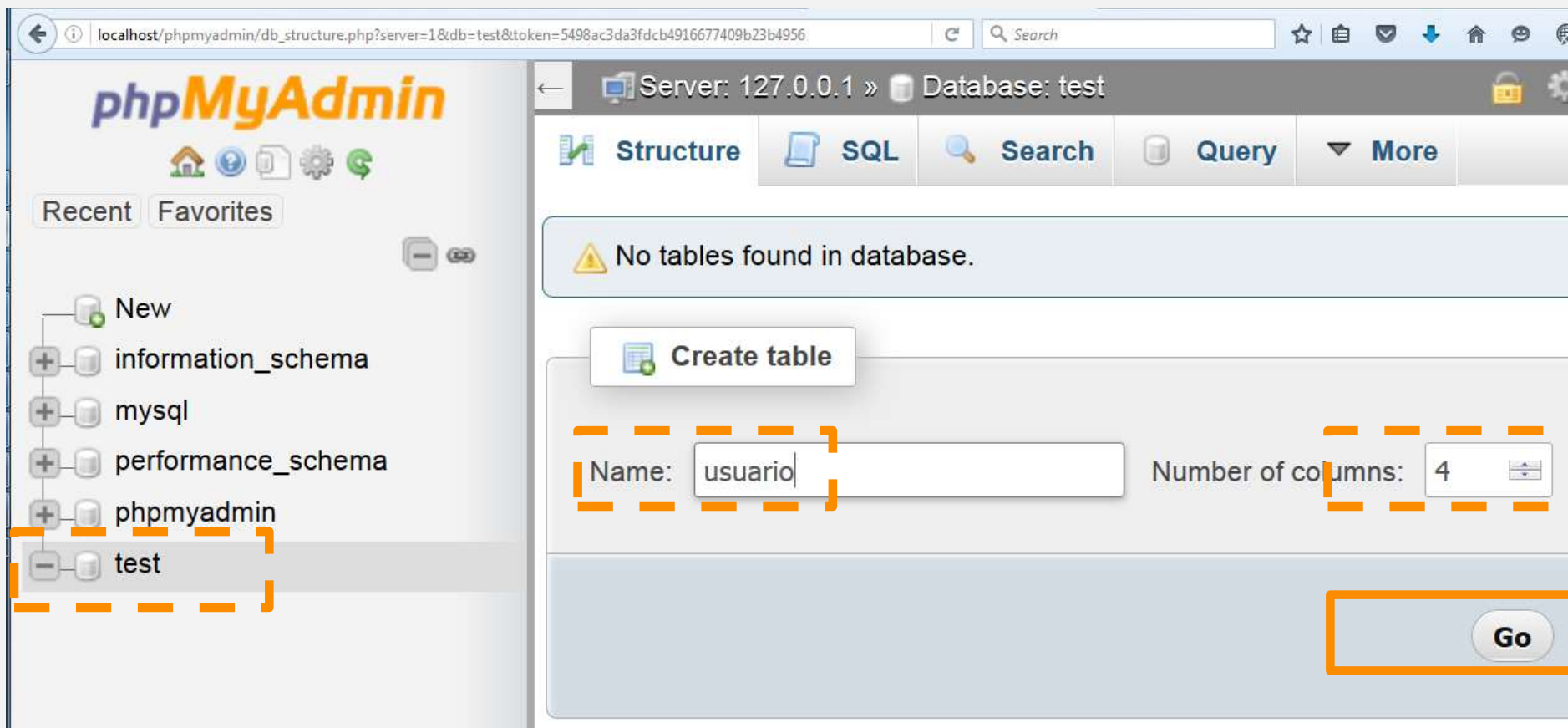
Exemplo de tabela do MySQL

Tabela *Pessoa*

id	nome	idade	profissão
0	João da Silva	28	Desenvolvedor(a)
1	Maria da Silva	27	Desenvolvedor(a)

Conexão com MySQL

- ✓ Para criar uma tabela no MySQL vamos utilizar o phpMyAdmin. Para acessá-lo, basta acessar a URL <http://localhost:8081/phpmyadmin>.
- ✓ A imagem abaixo mostra os pontos para criar uma tabela, com o nome de "usuario", com 4 colunas.



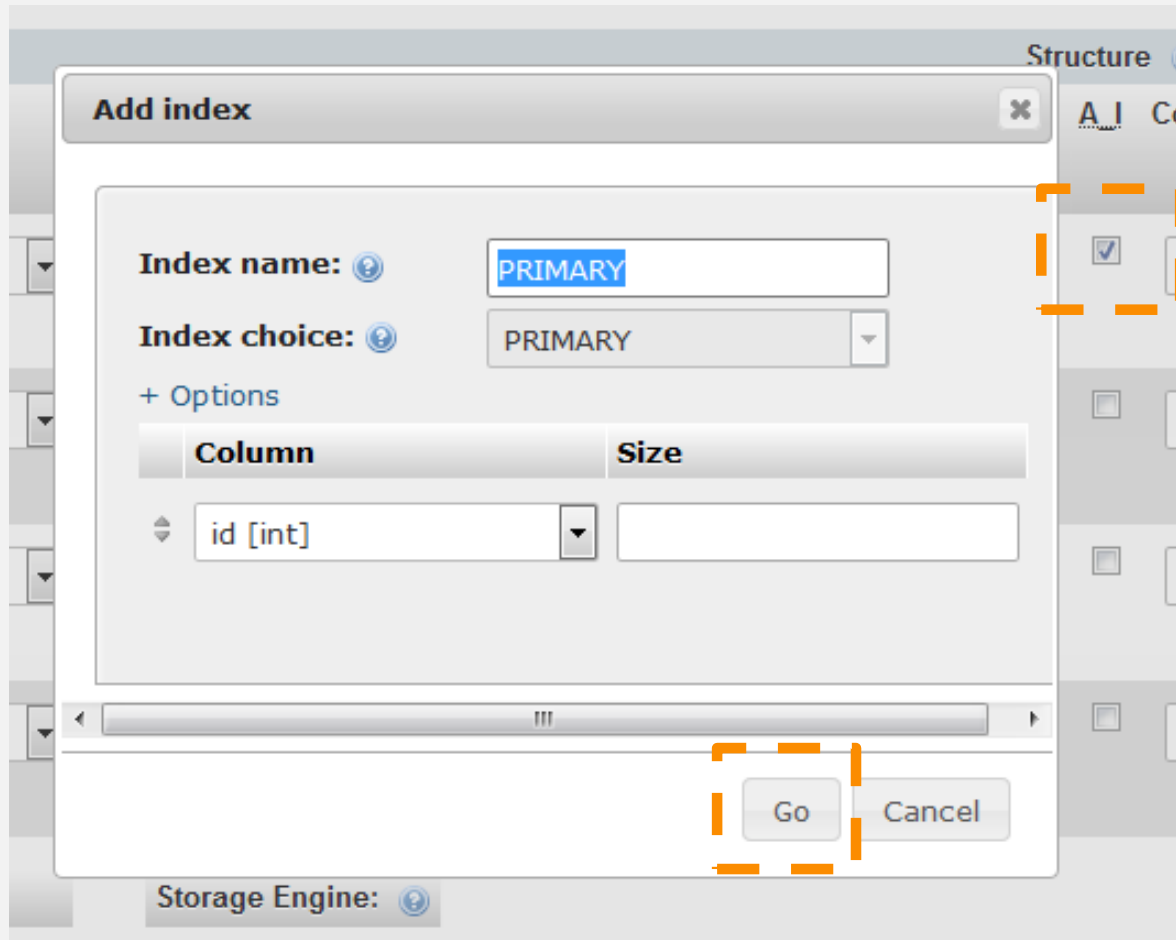
Conexão com MySQL

- ✓ As quatro colunas criadas deveram ser preenchidas conforme mostra a imagem abaixo.

Name	Type ?	Length/Values ?
<input type="text" value="id"/>	<input type="text" value="INT"/> ▼	<input type="text"/>
Pick from Central Columns		
<input type="text" value="usr"/>	<input type="text" value="VARCHAR"/> ▼	<input type="text" value="100"/>
Pick from Central Columns		
<input type="text" value="sen"/>	<input type="text" value="VARCHAR"/> ▼	<input type="text" value="100"/>
Pick from Central Columns		
<input type="text" value="nome"/>	<input type="text" value="VARCHAR"/> ▼	<input type="text" value="100"/>
Pick from Central Columns		

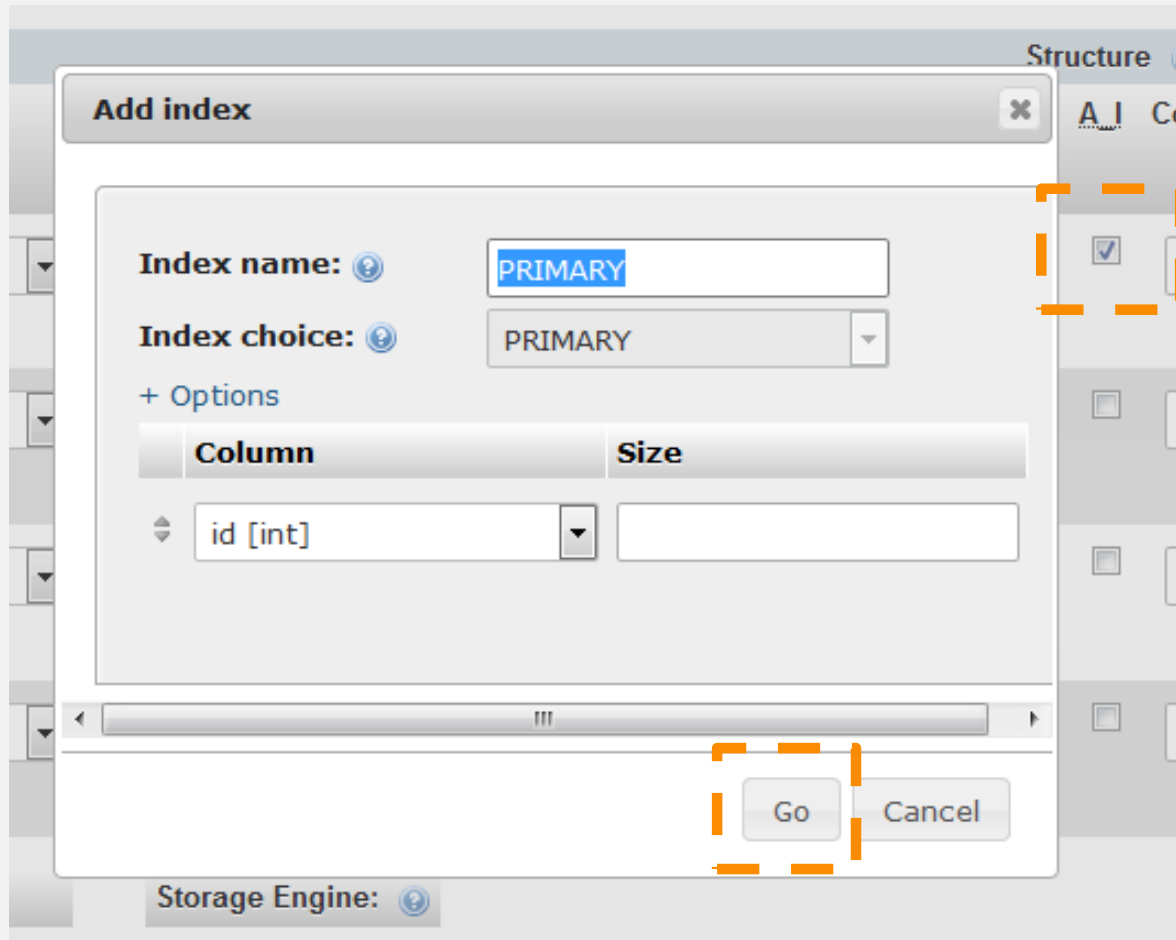
Conexão com MySQL

- ✓ As quatro colunas criadas deveram ser preenchidas conforme mostra a imagem abaixo.



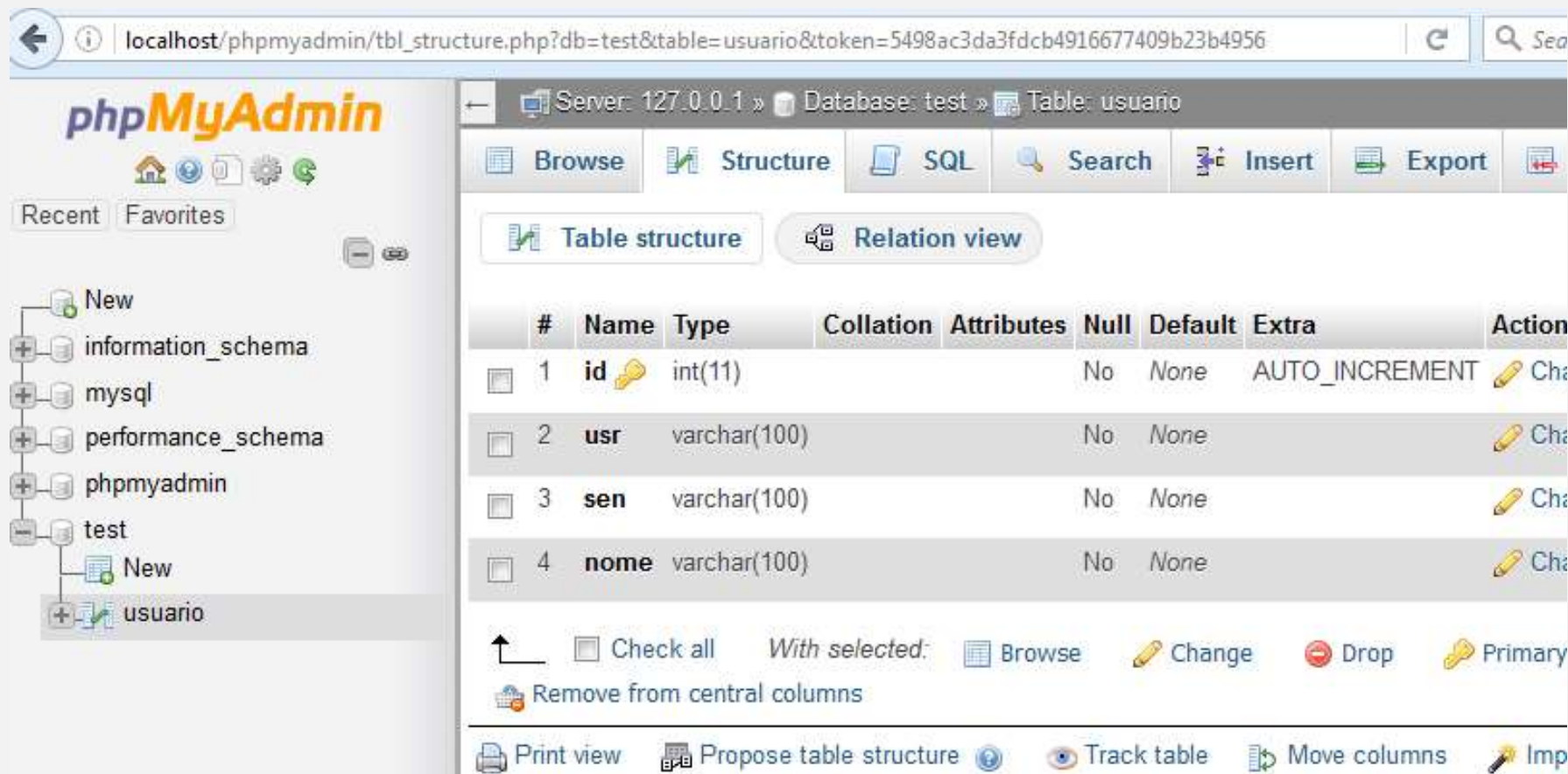
Conexão com MySQL

- ✓ Depois de marcar a coluna A_I (Auto Increment) da coluna "id", a caixa abaixo irá aparecer. Basta clicar no botão "Go".



Conexão com MySQL

- ✓ Depois de configurar o campo (coluna) "id" como A_I, basta clicar no botão "Save" no final/meio da tela na direita.
- ✓ A página será redirecionada mostrando a estrutura da tabela, como mostra a imagem abaixo



The screenshot shows the phpMyAdmin interface. The left sidebar displays the database structure with 'test' selected and 'usuario' highlighted. The main panel shows the 'Table structure' view for the 'usuario' table. The table has four columns: 'id' (int(11), primary key, auto-increment), 'usr' (varchar(100)), 'sen' (varchar(100)), and 'nome' (varchar(100)). The interface includes navigation tabs like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', and 'Export'. At the bottom, there are options to 'Print view', 'Propose table structure', 'Track table', 'Move columns', and 'Imp'.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Cha
2	usr	varchar(100)			No	None		Cha
3	sen	varchar(100)			No	None		Cha
4	nome	varchar(100)			No	None		Cha

Conexão com MySQL

conexao_bd.php

```
01 <?php
02     $conexao = mysqli_connect("localhost:3308", "root", "", "test");
03     if (!$conexao)
04         die("Erro na conexão com MySQL");
05 ?>
06
07
08
09
10
11
12
13
14
15
16
17
```

Criação de um sistema de cadastro e *login*

Criação de um sistema de cadastro e login

cadastro.php

```
01 <h1>Cadastro</h1>
02 <form action="cadastro2.php" method="post">
03     <label for="usuario">Usuário</label>
04     <br>
05     <input type="text" name="usuario" id="usuario">
06     <br>
07     <label for="nome">Nome</label>
08     <br>
09     <input type="text" name="nome" id="nome">
10     <br>
11     <label for="senha">Senha</label>
12     <br>
13     <input type="text" name="senha" id="senha">
14     <br>
15     <input type="submit" value="Cadastrar">
16 </form>
17
```

Criação de um sistema de cadastro e login

cadastro2.php

```
01 <?php
02     require("conexao_bd.php");
03     $usuario = $_POST["usuario"];
04     $senha = $_POST["senha"];
05     $nome = $_POST["nome"];
06     $sql = "INSERT INTO usuario (usr, sen, nome) VALUES ('" . $usuario . "', '" . $senha
07     . "', '" . $nome . "')";
08     if (mysqli_query($conexao, $sql))
09         echo "Cadastrado com sucesso";
10     else
11         echo "Houve um erro";
12
13
14
15
16
```

Criação de um sistema de cadastro e login

login.php

```
01 <h1>Login</h1>
02 <form action="login2.php" method="post">
03     <label for="usuario">Usuário</label>
04     <br>
05     <input type="text" name="usuario" id="usuario">
06     <br>
07     <label for="senha">Senha</label>
08     <br>
09     <input type="text" name="senha" id="senha">
10     <br>
11     <input type="submit" value="Entrar">
12 </form>
13
14
15
16
17
```

Criação de um sistema de cadastro e login

login2.php

```
01 <?php
02     require("conexao_bd.php");
03     session_start();
04     $usuario = $_POST["usuario"];
05     $senha = $_POST["senha"];
06     $sql = "SELECT * FROM usuario WHERE usr = '" . $usuario . "' AND sen = '" . $senha
07     . "'";
08     $result = mysqli_query($conexao, $sql);
09     if (mysqli_num_rows($result) > 0) {
10         $row = mysqli_fetch_array($result);
11         $_SESSION["usuario_logado"] = $row;
12         echo "Bem vindo - " . $_SESSION["usuario_logado"]["nome"];
13     } else {
14         echo "Usuário ou senha incorretos";
15     }
16
```

Criação de um sistema de cadastro e login

login.php

```
01 <h1>Login</h1>
02 <?php
03     session_start();
04     if ($_SESSION["usuario_logado"]) {
05         ?>
06         Você já está logado como <?=$_SESSION["usuario_logado"]["nome"]?><br>
07         <button onclick="location.href = 'sair.php'">Deslogar</button>
08         <?php
09     } else {
10         ?>
11         <form action="login2.php" method="post">
12             [...]
13         </form>
14         <?php
15     }
16 ?>
17
```


Criação de um sistema de cadastro e login

sair.php

```
01 <?php
02     session_start();
03     session_destroy();
04     echo "Deslogado";
05
06
07
08
09
10
11
12
13
14
15
16
17
```



Obrigado!

Otávio R. Rossi
Guilherme Uezima

Contatos



Guilherme Uezima

Design e Integração

+ 55 11 9 7692-7785

guilherme@inopus.com.br



Gustavo Soré

Desenvolvimento Back-end

+ 55 11 9 8585-4328

gustavo@inopus.com.br



Michel Zarzour Filho

Desenvolvimento Front-end

+ 55 11 9 8174-7375

michel@inopus.com.br



Otávio R. Rossi

Consultor de Negócios

+ 55 11 9 8544-4724

otavio@inopus.com.br

Referências

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form>
- <https://www.shimmercat.com/en/blog/articles/whats-push/>
- <http://php.net/manual/en/function.mysql-connect.php>
- <http://php.net/manual/en/mysql.close.php>
- <http://php.net/manual/en/mysql.error.php>
- <https://tools.ietf.org/html/rfc7540>
- <https://developer.mozilla.org/en/docs/AJAX>
- [https://msdn.microsoft.com/en-us/library/ms526971\(v=exchg.10\).aspx](https://msdn.microsoft.com/en-us/library/ms526971(v=exchg.10).aspx)
- [https://msdn.microsoft.com/en-us/library/ms527355\(v=exchg.10\).aspx](https://msdn.microsoft.com/en-us/library/ms527355(v=exchg.10).aspx)
- <http://db-engines.com/en/ranking/relational+dbms>
- <http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- <https://www.phpmyadmin.net/about/>