

Centro Paula Souza
ETEC Dr Geraldo José Rodrigues Alckmin

Apostila de PHP e MySQL

Versão 1.0

Componente Curricular: Programação WEB
Série atendida: 2^a e 3^a

Autores: Gilberto Abud Junior
Reginaldo Luiz Gonçalves

2023



Diagramação

Gilberto Abud Junior
Reginaldo Luiz Gonçalves

Design da capa e projeto gráfico

Gilberto Abud Junior
Reginaldo Luiz Gonçalves

Imagem da capa

Gilberto Abud Junior
Reginaldo Luiz Gonçalves

Revisão de texto

Gilberto Abud Junior
Reginaldo Luiz Gonçalves

© 2023 Edição brasileira
by Home Editora

© 2023 Texto
by Home Editora
Todos os direitos reservados

Home Editora

CNPJ: 39.242.488/0002-80

www.homeeditora.com

contato@homeeditora.com

9198473-5110

Av. Augusto Montenegro, 4120 - Parque Verde, Belém - PA,
66635-110

Editor-Chefe

Prof. Dr. Ednilson Ramalho

Revisão, diagramação e capa

Autor

Produtor editorial

Laiane Borges

Catálogo na publicação

Elaborada por Bibliotecária Janaina Ramos – CRB-8/9166

A165a

Abud Junior, Gilberto

Apostila de PHP e MySQL - Versão 1.0 / Gilberto Abud Junior, Reginaldo Luiz
Gonçalves. – Belém: Home, 2023.

50 p., fotos.; 16 X 23 cm

Livro em pdf

ISBN: 978-65-85712-28-6

DOI: 10.46898/home.2adf09b5-f7cc-4eb1-a049-bb6c14a72a46

I. Banco de dados. I. Abud Junior, Gilberto. II. Gonçalves, Reginaldo Luiz. III.
Título.

CDD 005.75

Índice para catálogo sistemático

I. Banco de dados

Ficha de Identificação de Material Didático e Autor

1. Título: **Apostila de PHP e MySQL – Versão 1.0.**
2. Assunto: **Técnicas e Linguagem de Programação para Internet.**
3. Resumo: **Além da explicação teórica rápida, a apostila avança na utilização da construção de Sites com conexão por meio do PHP e MySQL, utilizando o XAMPP, com vários exemplos práticos desenvolvidos em sala de aula com os alunos.**
4. Área / habilitação a que se destina:
Área/Eixo – Informática e Comunicação.
Habilitações: Desenvolvimento de Sistemas.
5. Componentes Curriculares que atinge:
Programação WEB II e III.
6. Séries atendidas: **2ª e 3ª.**
7. Nome do (s) autor (es): **Gilberto Abud Junior**
Reginaldo Luiz Gonçalves
8. Unidade Escolar dos autores:
Etec Dr. Geraldo José Rodrigues Alckmin – Etec Taubaté.
9. Telefones de contato dos autores: **(12) 99104-9082**
(12) 99108-3878
10. E-mail dos autores:
gilberto.junior101@etec.sp.gov.br
reginaldo.goncalves@etec.sp.gov.br
11. Número de páginas: 50, caracteres com espaço: 27058, laudas: 19.

Sumário

Os autores	6
Agradecimentos	7
Apresentação	8
Introdução	9
Capítulo 01	10
PHP com Banco de dados.....	10
Capítulo 02	13
Site de Pesquisa.....	13
Capítulo 03	22
Exportar e Importar em SQL.....	22
Capítulo 04	25
Banco de Dados Idiomas	25
Apêndice 01.....	39
PHP – Conceitos Básicos.....	39
Tipos de Dados	41
Variável.....	42
Operadores Aritméticos	44
Operadores Relacionais.....	45
Operadores Lógicos.....	45
Referências	47
Comentários finais.....	49

Os autores

Gilberto Abud Junior

Brasileiro, nascido em Taubaté - SP, é graduado Tecnólogo em Processamento de Dados, Pedagogia e Matemática. Possui Formação Pedagógica para Educação Profissional em Nível Médio. É Especialista em Informática em Educação e Administração Escolar. Atua como professor no curso Técnico em Informática desde 1998 no Colégio UNITAU da Universidade de Taubaté e desde 2000 nas unidades das Etecs do Centro Paula Souza, atualmente na unidade de Taubaté - SP.

Reginaldo Luiz Gonçalves

Brasileiro, nascido em Taubaté - SP, é graduado em Computação Científica e Pedagogia. Possui Formação Pedagógica para Educação Profissional em Nível Médio. É Especialista em Informática em Educação, Administração Escolar, Educação a Distância e em Currículo, Didática e Metodologias Ativas. Atua como professor no curso Técnico em Informática desde 1994 no Colégio UNITAU da Universidade de Taubaté e desde 2003 nas unidades das Etecs do Centro Paula Souza, atualmente na unidade de Taubaté - SP.

Agradecimentos

Aos nossos familiares, pela paciência, pelo incentivo e pelo apoio incondicional nos momentos difíceis, principalmente nos momentos de nossa ausência para a dedicação a esse trabalho.

Apresentação

Olá,

Caros estudantes, a programação é uma ciência que utiliza a lógica de nosso pensamento para resolvermos problemas dos mais variados níveis. Com ela conseguimos transmitir e “ensinar” ao computador o que ele deve fazer para a solução do problema proposto. Aprendendo a lógica com certeza você estará habilitado a programar em qualquer linguagem que se interessar.

Esta apostila tem o objetivo de atender uma demanda com mais exemplos práticos no que diz respeito a construção de Sites com conexão por meio do PHP e MySQL, utilizando o XAMPP, que precisam ser bem entendidos. Além da explicação teórica rápida, acompanha alguns exemplos práticos desenvolvidos em sala de aula com os alunos dos Cursos Técnicos em Informática, Informática para Internet e Desenvolvimento de Sistemas. A metodologia de resolução de problemas é aplicada e cada atividade aqui resolvida foi estudada, explicada e construída pelo professor e pelos alunos dos respectivos cursos.

Bons estudos aos leitores!

Introdução

Analisar nossa maneira de pensar nos leva a melhorar o pensamento lógico. As linguagens de programação servem para isso. Ao determinarmos o que desejamos que o computador faça, estamos “conversando” com ele. Os computadores, só fazem aquilo que mandamos, e não necessariamente o que desejamos que eles façam. Não deve haver nenhuma ambiguidade nas instruções dos programas que fornecemos ao computador, nem a possibilidade de interpretações alternativas.

O computador sempre tomará algum caminho em suas ações; muito cuidado é necessário para assegurar que o computador siga pelo único caminho correto possível que leve aos resultados desejados. Quando o aluno está interagindo com o computador ele está manipulando conceitos e isso contribui para seu desenvolvimento mental. Ele está adquirindo conceitos da mesma maneira que ele adquire conceitos quando interage com objetos do mundo.

Os alunos aprendem, porque essa interação com o computador propicia um ambiente riquíssimo e bastante efetivo do ponto de vista de construção do conhecimento.

Desejamos aos leitores um bom estudo e sucesso na programação e na construção do conhecimento.

Capítulo 01

PHP com Banco de dados

Um dos recursos muito interessantes do PHP é sua facilidade de conexão com banco de dados. Podemos definir um banco de dados como um conjunto de dados organizados logicamente para atender um ou mais usuários. Eles são geridos por um software com funções que nos permitem definir, recuperar, inserir, alterar e excluir dados. A esse software damos o nome de Sistema Gerenciador de Banco de Dados. Utilizaremos nos exemplos desenvolvidos nessa Apostila o **MySQL**, que utiliza a linguagem **SQL - Structured Query Language** (Linguagem de Consulta Estruturada), uma linguagem padrão para o trabalho com banco de dados. Utilizaremos o pacote **XAMPP** que pode ser baixado no link abaixo.

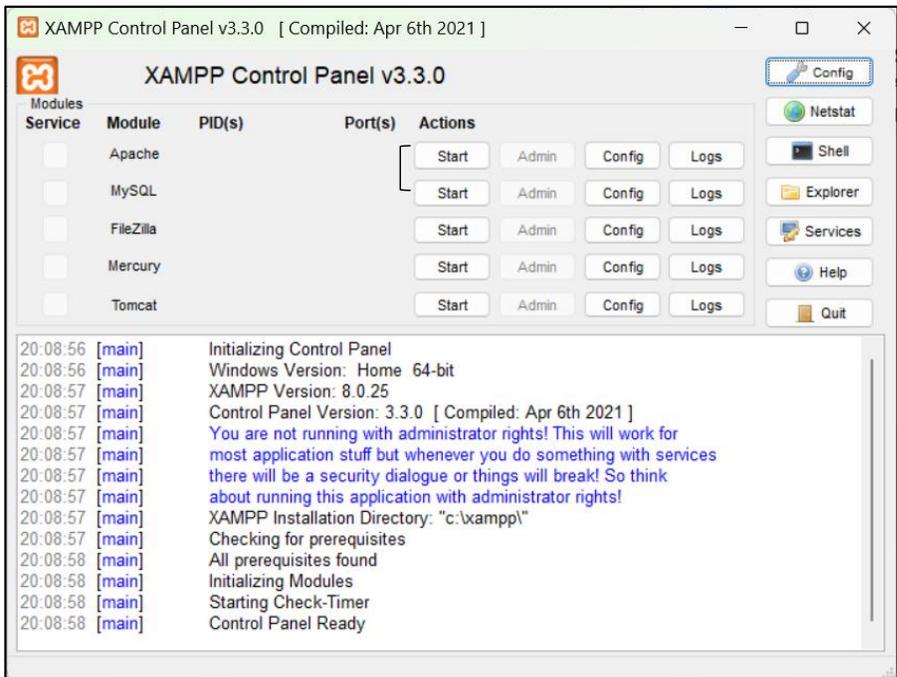
https://www.apachefriends.org/pt_br/index.html

Ao instalar o **XAMPP** conseguiremos utilizar o **MySQL** e todos os seus recursos por meio do **phpMyAdmin**, que permite a administração do **MySQL**. Com ele podemos criar e apagar bancos de dados, inserir, atualizar, deletar tabelas de maneira fácil e rápida.

Após a instalação do **XAMPP** devemos abri-lo na opção **XAMPP, Control Panel**.



Uma janela será aberta e devemos iniciar (Start) nas opções **Apache** e **MySQL**:



As duas opções ficarão assinaladas, permitindo o acesso por meio da opção Admin. Para criarmos os bancos de dados utilizados nos exemplos nos próximos capítulos acionaremos o Admin do **MySQL**.

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	16296 15764	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	13392	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

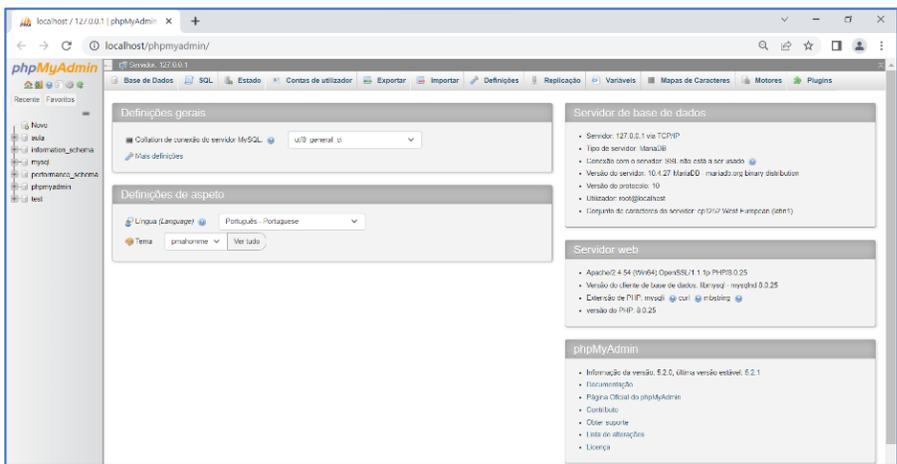
20:08:56 [main] Initializing Control Panel
 20:08:56 [main] Windows Version: Home 64-bit
 20:08:57 [main] XAMPP Version: 8.0.25
 20:08:57 [main] Control Panel Version: 3.3.0 [Compiled: Apr 6th 2021]
 20:08:57 [main] You are not running with administrator rights! This will work for most application stuff but whenever you do something with services there will be a security dialogue or things will break! So think about running this application with administrator rights!
 20:08:57 [main] XAMPP Installation Directory: "c:\xampp\
 20:08:57 [main] Checking for prerequisites
 20:08:58 [main] All prerequisites found
 20:08:58 [main] Initializing Modules
 20:08:58 [main] Starting Check-Timer
 20:08:58 [main] Control Panel Ready
 20:12:17 [Apache] Attempting to start Apache app...
 20:12:17 [Apache] Status change detected: running
 20:12:30 [mysql] Attempting to start MySQL app...
 20:12:30 [mysql] Status change detected: running

Acione o Admin do MySQL.

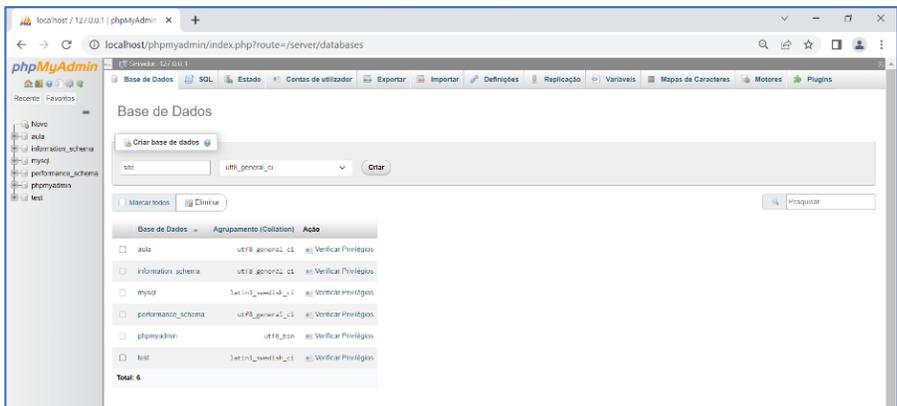
Capítulo 02

Site de Pesquisa

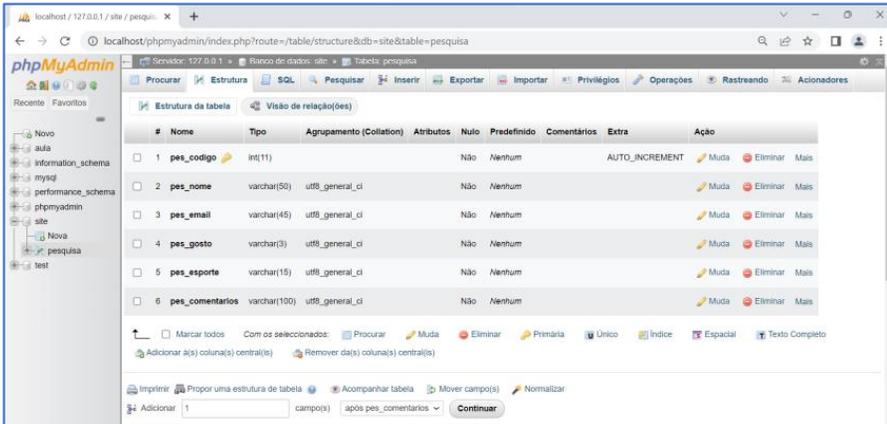
Iniciando a sequência do aprendizado, acione o Admin do MySQL, ao ser acionado, utilize o **phpMyAdmin**:



Em seguida crie um Banco de Dados com o nome **Site**:



Em seguida construa a **Tabela Pesquisa** com os seguintes atributos e propriedades, de acordo com a imagem abaixo:



Tela do **phpMyAdmin** com a estrutura da Tabela Pesquisa

Perceba que é utilizado no nome de cada campo na tabela a **Trigramação**, que consiste, de maneira simples, em iniciar com as três primeiras letras do nome da Tabela Pesquisa, neste caso, **pes**, método muito útil para ser utilizado ao fazer referências com outras Tabelas, por meio das *Chaves Estrangeiras*, que será visto com mais detalhes no Capítulo 3 desta Apostila. O campo **pes_codigo** é *Chave Primária* e autoincremento.

Em seguida acione o **Sublime Text** para dar início a construção do código **HTML** necessário para criar uma interface de comunicação por meio de um navegador, será utilizado um código **HTML** simples contendo formatação básica **CSS**, para isso digite o código da próxima imagem.

```

1  <!DOCTYPE HTML>
2  <html lang="pt-br">
3  <head>
4      <meta charset="utf-8">
5      <title>Pesquisa</title>
6      <style>
7          body {
8              background-color: #772211;
9          }
10         h1 {
11             font-size: 40px;
12             text-align: center;
13         }
14         table{
15             text-align: center;
16             border: #000 solid 1px;
17             width:50%;
18             font-size:20px;
19             background-color:#060;
20             color:#FFF;
21         }
22     </style>
23 </head>
24 <body>
25 <h1>PESQUISA</H1>
26 <TABLE align="center">
27     <TD>
28         <form id="fPesquisa" name="fPesquisa" method="post" action="Pesquisa.php">
29             <label>Informe seu nome:</label><br>
30             <input type="text" name="f_Nome" size="50"
31                 title="Digite seu Nome" required autofocus><br><br>
32
33             <label>Informe seu e-mail:</label><br>
34             <input type="text" name="f_email" size="45"
35                 title="Digite seu e-mail"><br><br>
36
37             <label>Você gostou da Competição InterClasses?</label><br>
38             <input type="radio" name="f_gostou" value="SIM" checked>Sim<br>
39             <input type="radio" name="f_gostou" value="NÃO">Não<br><br>
40
41             <label>Qual é o esporte de que mais gosta?</label>
42             <select name="f_Opcoes">
43                 <option value="Natação">Natação
44                 <option value="Futebol">Futebol
45                 <option value="Basquete">Basquete
46                 <option value="Volei" selected>Volei
47             </select>
48             <br><br>
49             <label>Deixe seus comentários:</label><br>
50             <textarea name="f_comentarios" rows=10 cols=45></textarea>
51             <br>
52             <input type="submit" value="Enviar" id="btn">
53             <input type="reset" value="Limpar" id="btn">
54         </form>
55     </td>
56 </table>
57 </body>
58 </html>

```

Arquivo pesquisa.html

Na pasta **htdocs** do XAMPP, crie uma pasta com o nome **SiteBD**, logo após o arquivo **pesquisa.html** deve ser gravado nela. Em seguida crie o arquivo **pesquisa.php** seguindo os códigos da imagem abaixo, grave este arquivo também na pasta **SiteBD**.

```
1 <?php
2 $con = mysqli_connect("localhost","root","");//conectar BD
3 mysqli_select_db($con,"site"); //selecionar banco de dados SITE
4
5 $res=mysqli_query($con,"SELECT * FROM PESQUISA");
6 $linhas=mysqli_num_rows($res);
7 echo "Encontrados $linhas registros na tabela PESQUISA <br>";
8
9 //Codigo=$_POST["f_Codigo"]; Sem necessidade por ser autoincremento
10 $Nome=utf8_decode($_POST["f_Nome"]);
11 $Email=$_POST["f_email"];
12 $Gostou=utf8_decode($_POST["f_gostou"]);
13 $Opcoes=utf8_decode($_POST["f_Opcoes"]);
14 $Comentarios=utf8_decode($_POST["f_comentarios"]);
15
16 $sql="INSERT INTO PESQUISA
17     VALUES (NULL,'$Nome','$Email','$Gostou','$Opcoes','$Comentarios)";
18 $res=mysqli_query($con,$sql);
19 $linhas=mysqli_affected_rows($con);
20
21 if($linhas == 1){
22     echo "Mais $linhas Registro gravado com sucesso <br>";
23 }else{
24     echo "Falha na gravação $linhas<br>";
25 }
26 mysqli_close($con);
27 ?>
28
29 <!doctype html>
30 <html lang="pt-br">
31     <head>
32         <title>Teste PHP com BD</title>
33         <meta charset="utf-8">
34         <script>
35             //Conteúdo Javascript
36         </script>
37     </head>
38 <body>
39     <!-- Conteúdo da página -->
40     <br><br>
41     <a href="Pesquisa.html">Voltar</a>
42 </body>
43 </html>
```

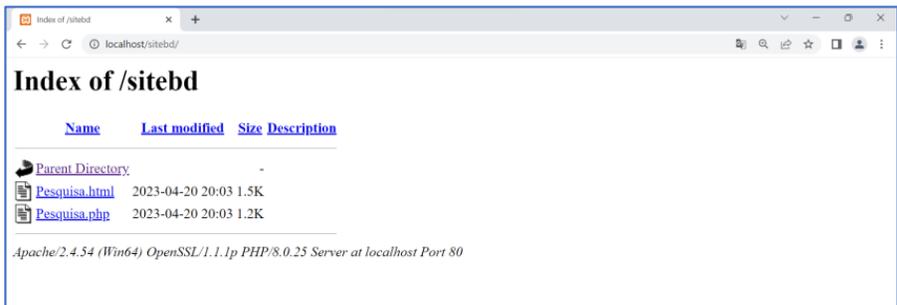
Arquivo pesquisa.php

Agora para darmos sequência a execução da nossa aplicação, vamos digitar **localhost** na barra de endereços do navegador, em seguida devemos acertar o caminho para os

arquivos que no caso basta apagar o **dashboard/** e substituí-lo por **SiteBD**.



Será apresentada a próxima tela:



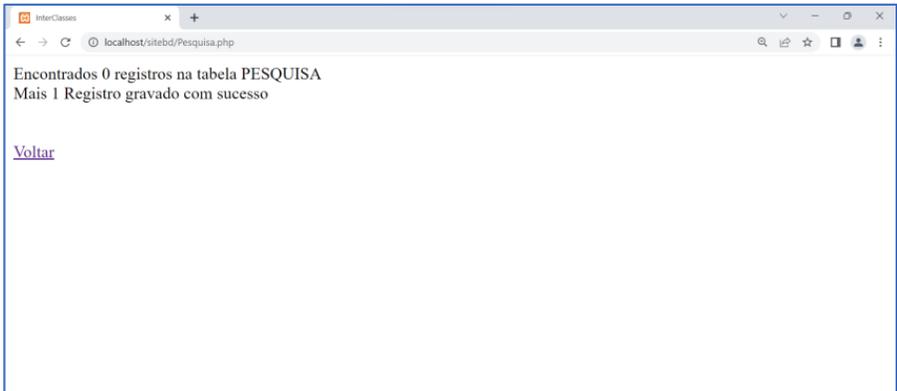
Acione o arquivo **Pesquisa.html** e como resultado temos a próxima tela:



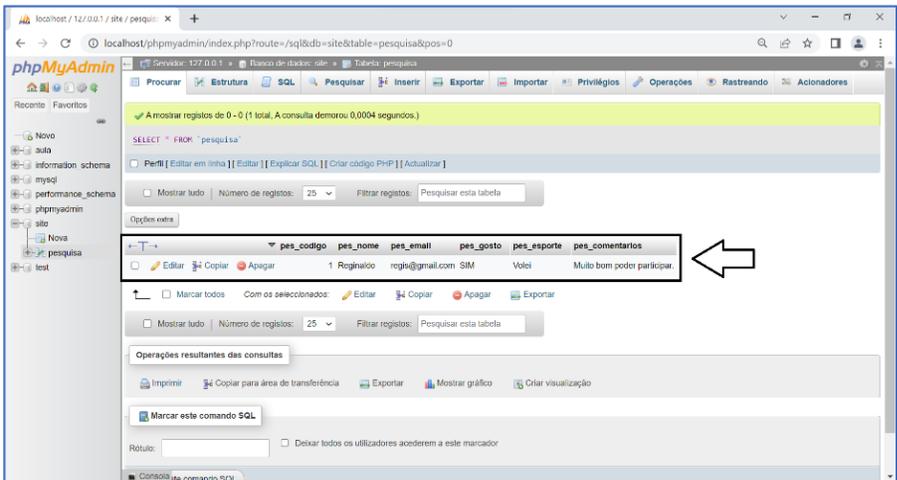
Informe os dados e clique no botão Enviar.



Após a digitação e clicar no botão Enviar, as seguintes informações serão apresentadas, veja a próxima tela.



Ao abrir o Banco de Dados **Site** por meio do **phpMyAdmin**, podemos selecionar a Tabela **Pesquisa** e verificar que os dados digitados no formulário, foram inseridos, como pode ser visto na próxima imagem.



Continuando com o teste, informe os seguintes dados e clique no botão Enviar.



As seguintes informações serão apresentadas, veja a próxima tela.



Ao clicarmos na opção **Procurar** no ambiente do **phpMyAdmin**, executamos uma consulta - `SELECT * FROM `pesquisa`` - e podemos verificar que os dados digitados no formulário, foram inseridos, como pode ser visto na próxima imagem.

The screenshot shows the phpMyAdmin interface for a database named 'pesquisa'. The table 'pesquisa' is selected, and the following SQL query is executed: `SELECT * FROM 'pesquisa'`. The table contains two records:

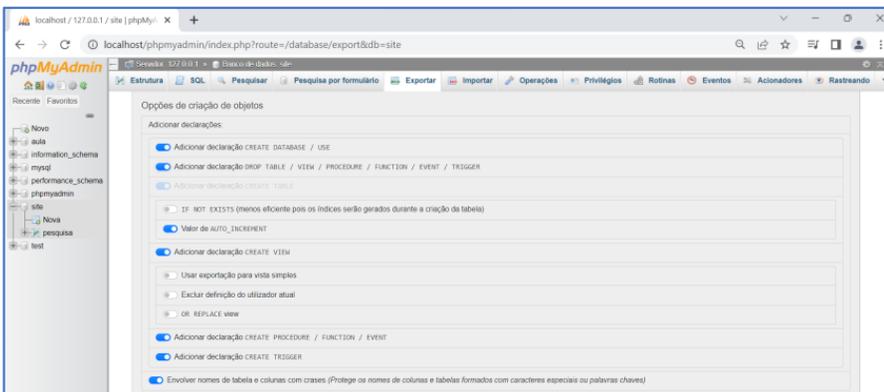
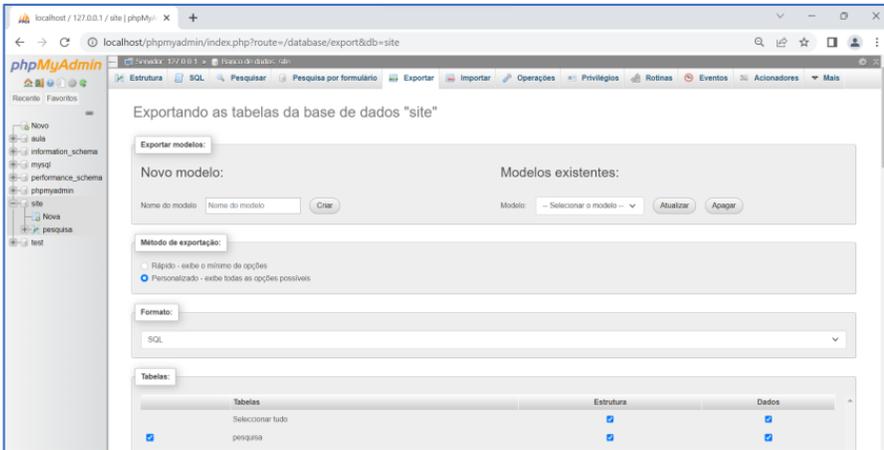
pes_codigo	pes_nome	pes_email	pes_gosto	pes_esporte	pes_comentarios
1	Reginaldo	regis@gmail.com	SIM	Volei	Muito bom poder participar.
2	Gilberto	gilberto@gmail.com	SIM	Futebol	Treino bastante para competir.

A white arrow points to the second record (Gilberto). Below the table, there are options to 'Operações resultantes das consultas' (Print, Copy, Export, Show graph, Create visualization) and a section to 'Marcar este comando SQL' (Mark this SQL command).

Capítulo 03

Exportar e Importar em SQL

O **phpMyAdmin** pode exportar o Banco de Dados para arquivos tipo texto, a exportação em **SQL** pode ser usada para restaurar seu banco de dados, portanto, é útil para **backup**, para isto, basta escolher a **guia “Exportar”**, fazer as configurações necessárias e clicar no **botão Exportar**. Será gerado o arquivo **site.sql**.



O arquivo gerado será do tipo texto. Abaixo é mostrado o conteúdo do exemplo utilizado:

```
-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Tempo de geração: 21-Abr-2023 às 01:55
-- Versão do servidor: 10.4.27-MariaDB
-- versão do PHP: 8.0.25

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Banco de dados: `site`
--
CREATE DATABASE IF NOT EXISTS `site` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
USE `site`;

-----

-- Estrutura da tabela `pesquisa`
--

DROP TABLE IF EXISTS `pesquisa`;
CREATE TABLE IF NOT EXISTS `pesquisa` (
  `pes_codigo` int(11) NOT NULL AUTO_INCREMENT,
  `pes_nome` varchar(50) NOT NULL,
  `pes_email` varchar(45) NOT NULL,
  `pes_gosto` varchar(3) NOT NULL,
  `pes_esporte` varchar(15) NOT NULL,
  `pes_comentarios` varchar(100) NOT NULL,
  PRIMARY KEY (`pes_codigo`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

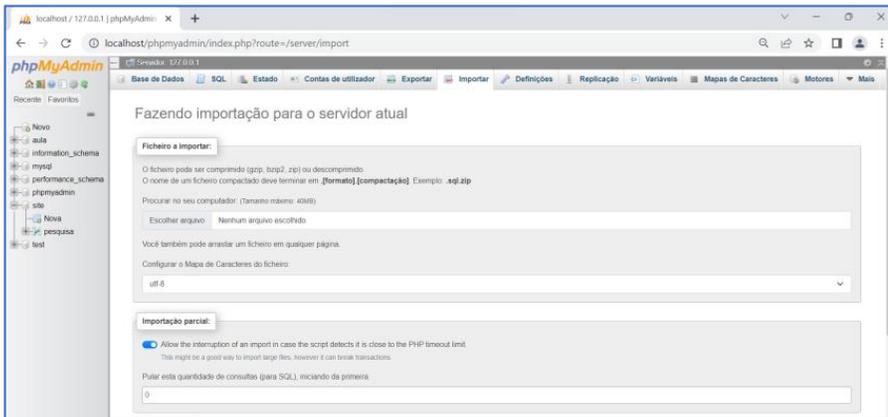
--
-- Extraindo dados da tabela `pesquisa`
--

INSERT INTO `pesquisa` (`pes_codigo`, `pes_nome`, `pes_email`, `pes_gosto`, `pes_esporte`, `pes_comentarios`) VALUES
(1, 'Reginaldo', 'regis@gmail.com', 'SIM', 'Volei', 'Muito bom poder participar.'),
(2, 'Gilberto', 'gilberto@gmail.com', 'SIM', 'Futebol', 'Treinei bastante para competir.');
```

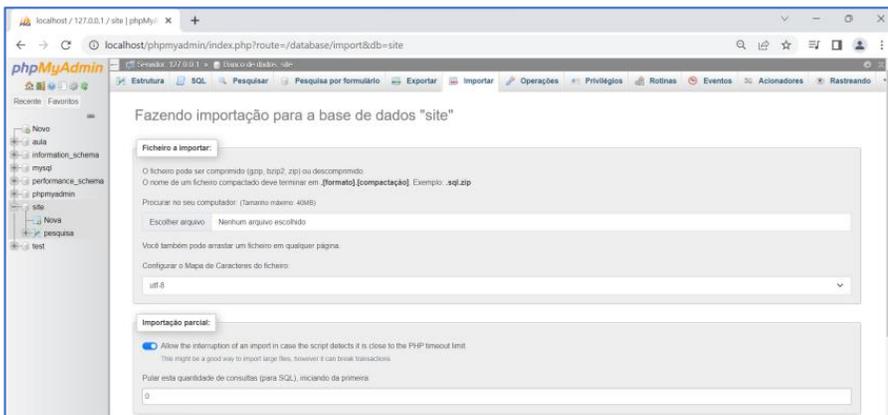
```
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@@OLD_COLLATION_CONNECTION */;
```

Para importar um Banco de Dados, vá para a guia “Importar” no phpMyAdmin.



Para importar **dados** para um Banco de Dados ou tabela específica, abra o Banco de Dados ou tabela antes de ir para a guia “Importar”.

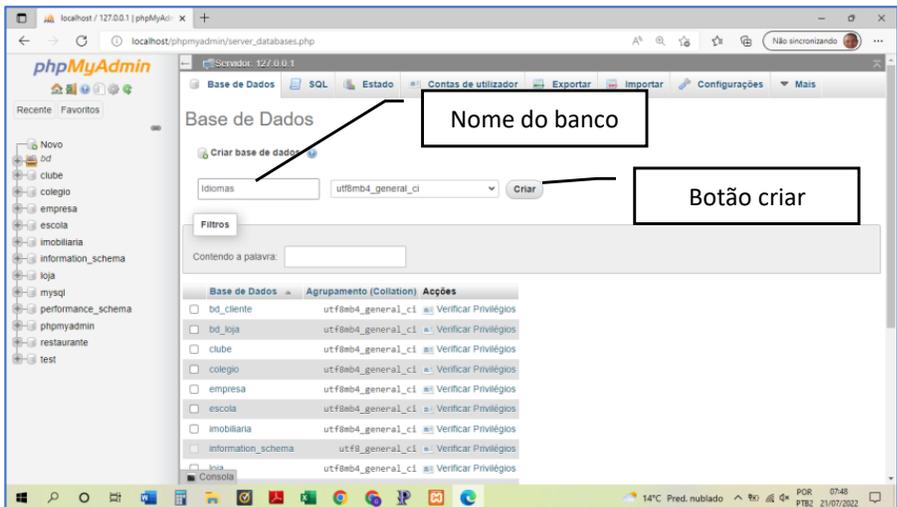


Além da guia padrão Importar e Exportar, você também pode importar um arquivo **SQL** diretamente arrastando e soltando-o de seu gerenciador de arquivos local para a interface **phpMyAdmin** em seu navegador da web.

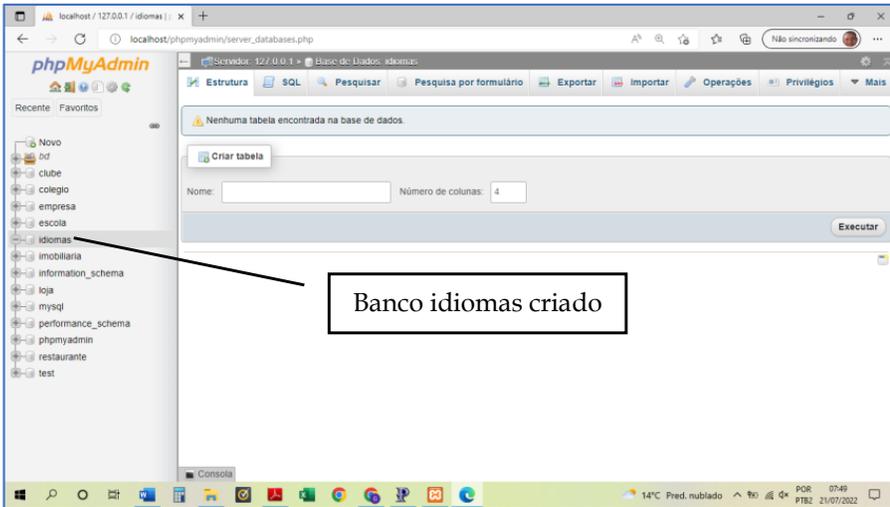
Capítulo 04

Banco de Dados Idiomas

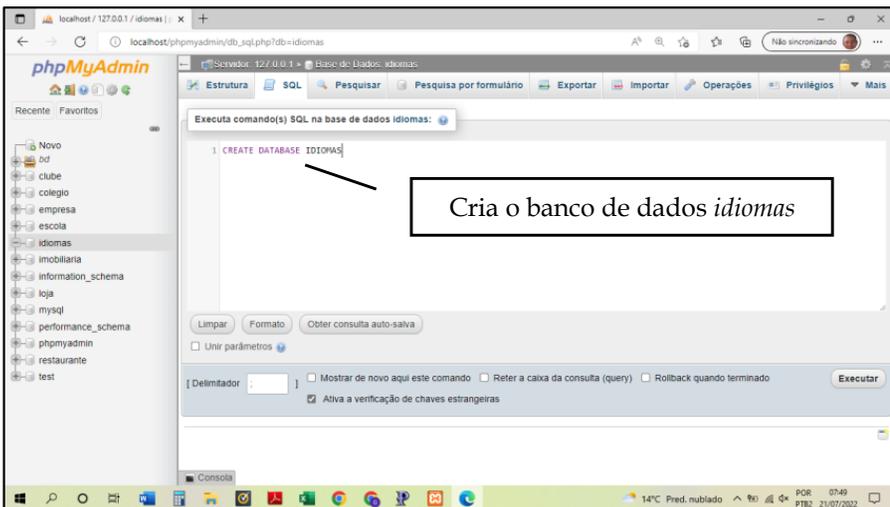
Pelo Admin do MySQL entraremos direto no **phpMyAdmin** para criarmos o banco de dados *idiomas* e duas tabelas chamadas *curso* e *aluno*. Na imagem abaixo inserimos na caixa de texto abaixo de Criar base de dados o nome do banco de dados que queremos criar que é *idiomas*. Depois clicamos no botão *Criar*.



Tela de criação do banco de dados idiomas.



Podemos criar o banco de outra maneira. Podemos usar a linguagem **SQL**, que por meio de comandos também nos permite criar um banco de dados e suas tabelas.



Tela de criação do banco de dados *idiomas* por comando SQL

Vamos agora criar as tabelas *curso* e *aluno* por meio dos comandos SQL.

Na linha 1 temos o comando CREATE TABLE CURSO, que criará a tabela *curso* para o nosso banco de dados.

Na linha 2 e 5 abrimos e fechamos parênteses que delimitam os campos que serão criados para a tabela.

Na linha 3 temos CUR_CODIGO INTEGER NOT NULL PRIMARY KEY, onde:

CUR_CODIGO é o nome do campo da tabela;

INTEGER é o tipo de dado que o campo receberá que será inteiro;

NOT NULL é um campo não nulo, que obrigatoriamente deve ser preenchido;

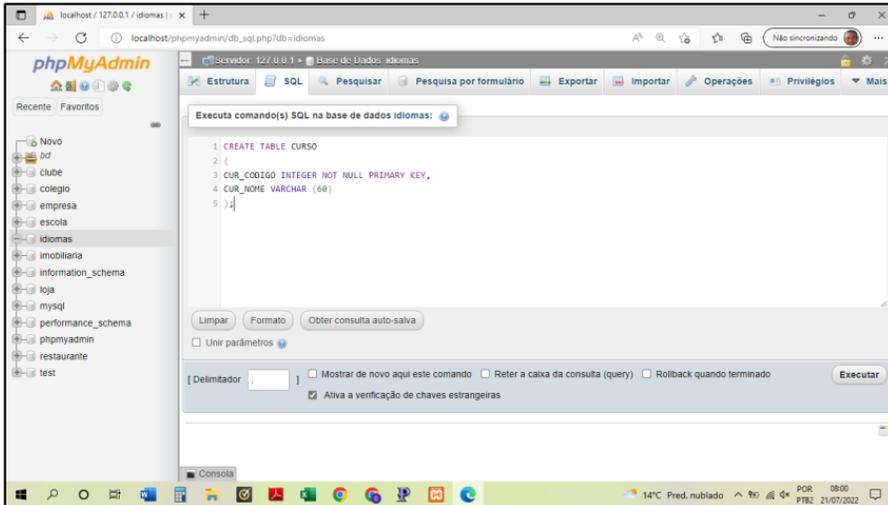
PRIMARY KEY diz que esse campo é a chave primária da tabela. Uma chave primária é um campo de uma tabela que é utilizado para identificar de forma única cada linha ou registro dessa mesma tabela. Ao final da linha colocamos uma vírgula que separa uma linha da outra ou um campo da tabela do outro;

Na linha 4 temos CUR_NOME VARCHAR (60), onde

CUR_NOME é o nome do campo;

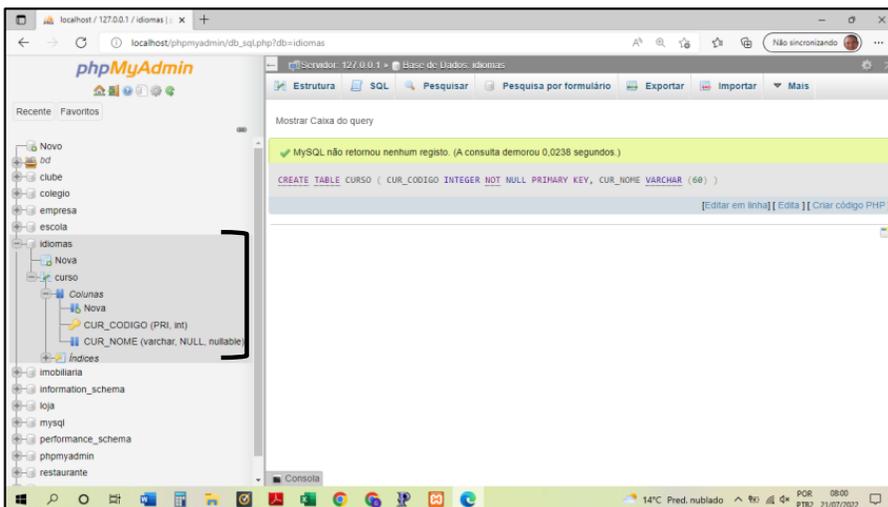
VARCHAR (60) é o tipo de dado alfanumérico com 60 caracteres para escrevermos o nome do curso. Repare que nessa linha não há vírgula ao final, pois o nome do curso é o último campo da tabela. Guarde então que na última linha de qualquer tabela criada, independente de quantos campos ela tiver, não

haverá vírgula, somente o parêntese já explicado anteriormente.



Tela dos comandos de criação da tabela *curso*.

Perceba que no lado esquerdo já aparece o banco de dados *idiomas* com a tabela *curso* e seus campos que acabamos de criar.



Tela exibindo a tabela *curso* criada.

O próximo passo é criar a tabela aluno para finalizarmos nosso banco de dados.

Na linha 1 temos `CREATE TABLE ALUNO` onde é criada a tabela com o nome que desejamos, no caso `ALUNO`;

Nas linhas 2 e 7 temos parênteses que delimitam os campos que serão criados em nossa tabela;

Na linha 3 temos `ALU_CODIGO INTEGER NOT NULL PRIMARY KEY`, onde:

`ALU_CODIGO` é o nome do campo;

`INTEGER` é o tipo de dado que será inserido;

`NOT NULL` define que este campo não pode ser nulo;

`PRIMARY KEY` define que esse campo é a chave primária desta tabela

Na linha 4 temos `ALU_NOME VARCHAR (60)` onde:

`ALU_NOME` é o nome do campo;

`VARCHAR(60)` é o tipo de dado alfanumérico com 60 caracteres (pode ser diferente de 60 sem problemas);

Nas linhas 5 e 6 faremos o relacionamento entre as tabelas `curso` e `aluno` definindo um campo como chave estrangeira (chave que permite a referência a registros que possuem origem em outras tabelas). Primeiramente criamos o campo. Utilizei aqui o mesmo campo que é chave primária na tabela `curso` que saindo de seu território será definida como chave estrangeira na tabela `aluno`. Ela também deverá ser do mesmo tipo do campo de origem. Por isso o comando foi escrito

dessa maneira: CUR_CODIGO INTEGER NOT NULL, ou seja, o campo CUR_CODIGO será inteiro e não nulo.

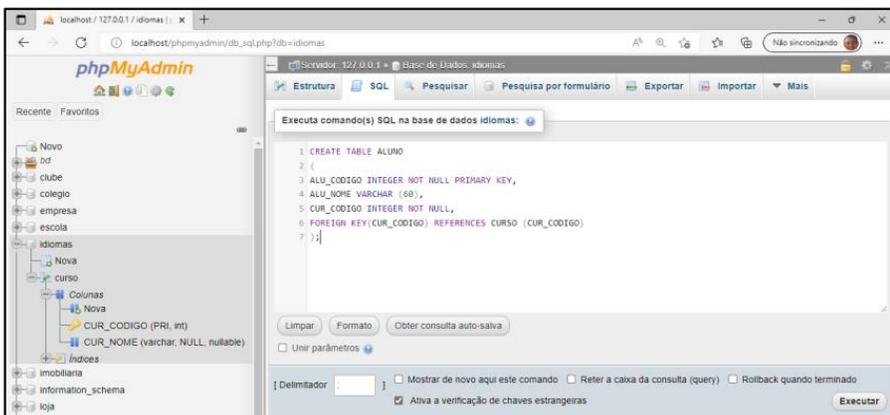
Agora com a linha FOREIGN KEY (CUR_CODIGO) REFERENCES CURSO (CUR_CODIGO) a chave estrangeira foi definida. Vejamos:

FOREIGN KEY (CUR_CODIGO) define que o campo CUR_CODIGO criado na tabela aluno será a chave estrangeira;

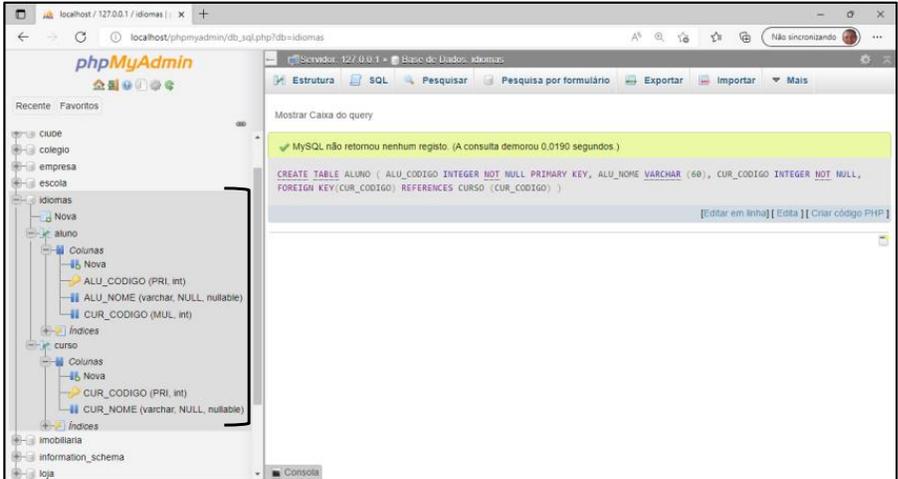
REFERENCES CURSO (CUR_CODIGO) significa que ele se relacionará com o CUR_CODIGO da tabela curso.

Sugiro aqui o livro “Projeto de Banco de Dados: Uma visão Prática” de Felipe Machado e Maurício Abreu, para um estudo aprofundado de banco de dados.

Seguindo a explicação perceba nas telas abaixo a escrita dos comandos SQL que criam a tabela aluno e como ficou depois de pronta. Agora podemos ver o banco criado, suas duas tabelas e os respectivos campos de cada tabela.

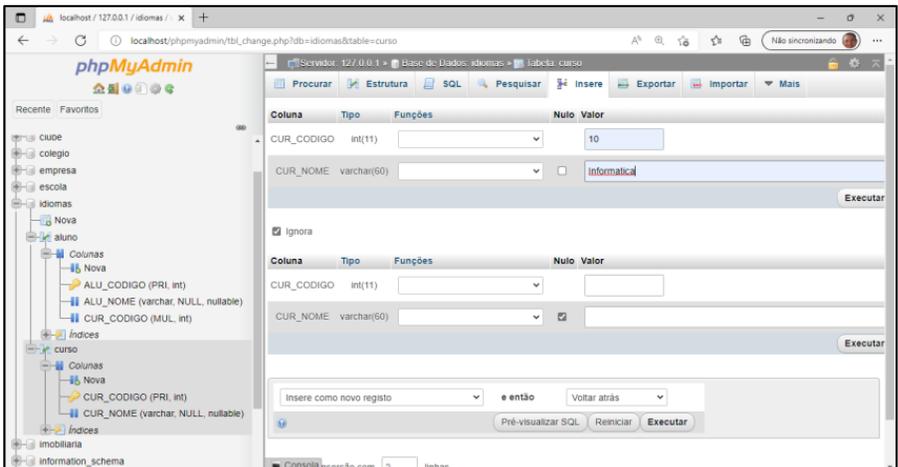


Tela de criação da tabela *aluno*.



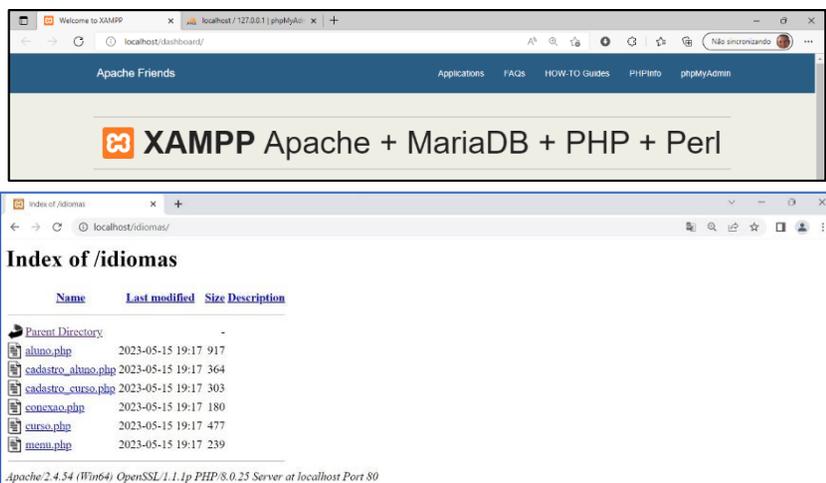
Tela de exibição do banco de dados completo.

Na guia **insere** do **phpMyAdmin** podemos inserir os dados que desejamos em qualquer uma das duas tabelas sem construir nenhuma página de internet, mas esse não é o nosso objetivo no momento. Então veremos a seguir como crio uma página que me permite inserir os dados para que eles sejam cadastrados no banco de dados de nossa escola de idiomas. Vamos acionar o Admin do Apache que já aprendemos:



Tela de cadastramento de dados.

Desenvolveremos 6 arquivos **php** para a solução dessa tarefa de cadastrar os dados do curso e dos alunos de nossa escola de idiomas. Devemos criar dentro da pasta **htdocs** do **XAMPP**, uma pasta chamada **idiomas**, os 6 arquivos **php** a seguir, deverão ser salvos dentro da pasta **idiomas**. Agora para darmos sequência a execução da nossa aplicação, vamos digitar **localhost** na barra de endereços do navegador, em seguida devemos acertar o caminho para os arquivos que neste caso basta apagar o **dashboard/** e substituí-lo por **idiomas**.



Código do arquivo de conexão.

```
1 <?php
2 $conexao = mysqli_connect("localhost","root","","idiomas");
3 if (!$conexao) {
4 die('Nao foi possivel conectar ao banco de dados.
5 Erro detectado: ' . mysqli_error());
6 }
7 ?>
8
```

Nas linhas 1 e 7 delimitamos o código php;
Na linha 2, uma variável \$conexao recebe o comando de conexão onde:
mysqli_connect é o comando MySQL;
localhost é o servidor local;
root é o usuário;
"" é onde colocamos a senha que nesse momento não o faremos e, idiomas é o nome do nosso banco

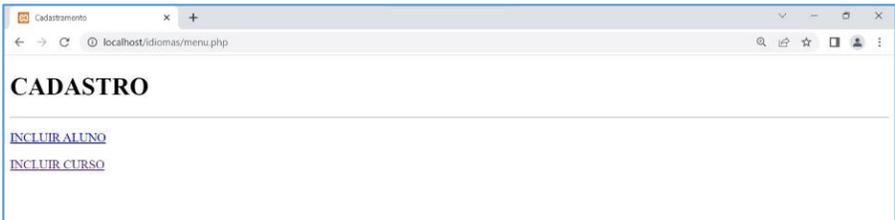
Arquivo conexao.php.

Código do Menu para navegação

Para uma melhor compreensão do HTML, faça uma pesquisa sobre os comandos na imagem abaixo.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Cadastramento</title>
5     <meta charset="utf-8">
6 </head>
7 <body>
8 <h1>CADASTRO</h1>
9 <hr>
10 <p><a href="aluno.php">INCLUIR ALUNO</a></p>
11 <p><a href="curso.php">INCLUIR CURSO</a></p>
12 </body>
13 </html>
```

Arquivo menu.php



Tela do menu de navegação.

Na sequência temos os códigos do formulário e do arquivo php que farão a inclusão dos dados no banco *idiomas*. O código do formulário para o usuário inserir os dados já é nosso conhecido. Também será um formulário padrão, pois nosso interesse nesse livro é aprendermos a lógica e os comandos que conectam o php com banco de dados. Sugiro aqui o livro HTML5 e CSS3 de Ana Laura Gomes e Richard Martelli da Editora SENAC.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Cadastro de Cursos</title>
5     <meta charset="utf-8">
6 </head>
7 <body>
8 <p>Preencher dados do Curso:</p>
9 <fieldset><legend>Cadastro de Cursos</legend>
10     <form method="post" action="cadastro_curso.php">
11         <label for="codigo">Código:</label>
12         <input type="text" id="codigo" name="codigo"><br>
13         <label for="nome">Nome:</label>
14         <input type="nome" id="nome" name="nome"><br>
15         <input type="reset" value="Limpar" name="limpar">
16         <input type="submit" value="Enviar" name="submit">
17     </form>
18 </fieldset>
19 <p><a href="menu.php">Voltar</a></p>
20 </body>

```

Arquivo curso.php

Como vimos nos programas anteriores os dados digitados no formulário serão transferidos para as variáveis do arquivo cadastro_curso.php para que ele faça a inclusão dos dados em nosso banco. Vamos então ver linha a linha após a transferência dos dados para as variáveis \$codigo e \$nome.

O comando *require_once* incluirá o arquivo conexão.php para que seja possível a conexão.

A linha 6 do nosso código armazenará na variável \$insserir o comando SQL que faz a inclusão dos dados. Ou seja, INSERT INTO curso, significa que é para inserir na tabela curso nos respectivos campos cur_codigo e cur_nome os valores (VALUES) que estão nas variáveis \$codigo e \$nome recebidas no formulário da nossa página.

O comando *mysqli_query* executa os conteúdos das variáveis \$conexao e \$insserir. O comando *mysqli_close* encerra a conexão com nosso banco.

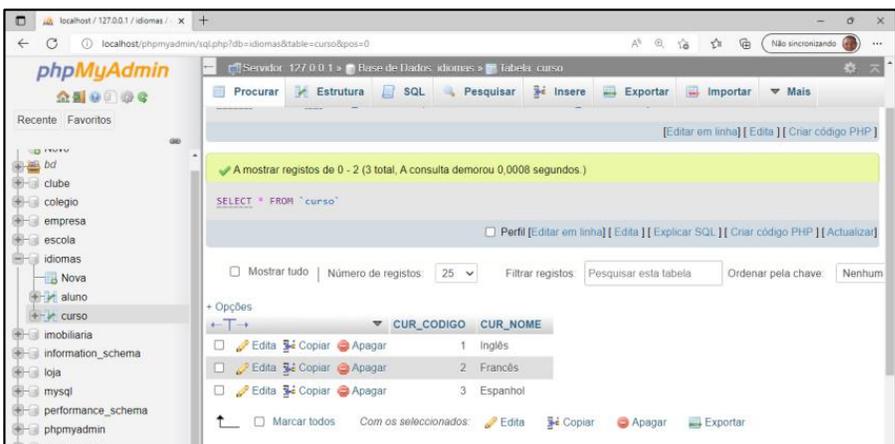
```
1 <?php
2     $codigo = $_POST['codigo'];
3     $nome = utf8_decode($_POST['nome']);
4     require_once("conexao.php");
5
6     $inserir = "INSERT INTO curso (cur_codigo, cur_nome) VALUES ('$codigo', '$nome)";
7
8     mysqli_query($conexao, $inserir);
9
10    mysqli_close($conexao);
11 ?>
12
13 <p><a href="curso.php">Voltar</a></p>
```

Arquivo cadastro_curso.php.

Veja como ficou o resultado da tela para o usuário e depois a imagem do banco de dados já com dados cadastrados da tabela curso.



Tela do site para cadastramento dos cursos



Tela dos cursos cadastrados no banco *idiomas*

Irá acontecer o mesmo com o cadastro dos dados dos alunos. Vamos ver que o código é muito parecido, com um detalhe importante que é o cadastramento utilizando o relacionamento entre as tabelas (chave estrangeira). Vejamos então o código do formulário e o de inserção de dados a seguir.

Perceba que além dos comandos HTML já estudados no formulário de cadastro de cursos, temos uma novidade: alguns comandos PHP em destaque que nos ajudarão por meio de um select, nos mostrar os cursos disponíveis para matrícula. Já no formulário de entrada dos dados é feita uma conexão com o banco, selecionando os campos da tabela curso para que seja escolhido o curso desejado pelo nome do curso, embora seja armazenado na tabela o código do curso que é a chave estrangeira.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5 </head>
6 <body>
7   <p>Preencher dados do aluno:</p>
8   <fieldset><legend>Cadastro de alunos</legend>
9   <form method="post" action="cadastro_aluno.php">
10    <label for="codigo">Código:</label>
11    <input type="text" id="codigo" name="codigo"><br>
12    <label for="aluno">Nome:</label>
13    <input type="text" id="aluno" name="aluno"><br>
14    <label for="curso">Selecione o Curso:</label>
15    <select name="curso" id="select">
16      <?php
17        require_once("conexao.php");
18        $consulta = mysqli_query($conexao, "SELECT cur_codigo, cur_nome FROM curso");
19        while ($dados = mysqli_fetch_array($consulta)) {
20          echo ("<option value='".$dados['cur_codigo']."'>".$dados['cur_nome']</option>");
21        }
22      <?>
23    </select><br>
24    <input type="reset" value="Limpar" name="limpar">
25    <input type="submit" value="Enviar" name="submit">
26  </fieldset>
27 </form>
28 <p><a href="menu.php">Voltar</a></p>
29 </body>
30 </html>
```

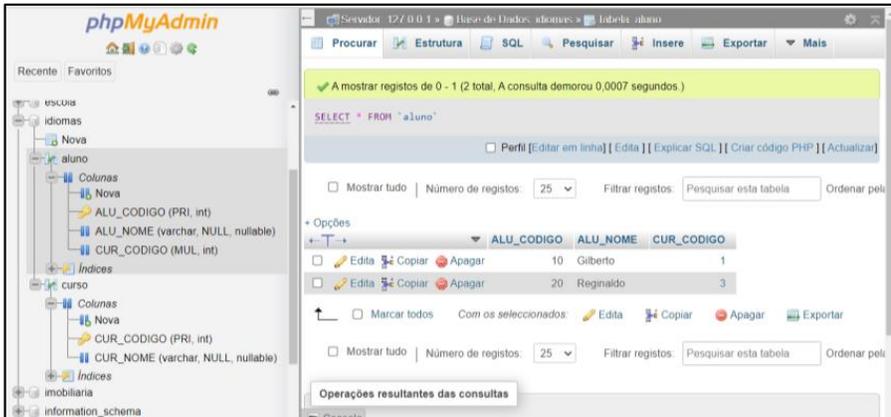
Arquivo aluno.php

```
1 <?php
2     $codigo = $_POST['codigo'];
3     $nome = $_POST['aluno'];
4     $curso = $_POST['curso'];
5     require_once("conexao.php");
6
7
8     $inserir = "INSERT INTO aluno ( alu_codigo, alu_nome, cur_codigo)
9     VALUES ('$codigo', '$nome', $curso) ";
10
11     mysqli_query($conexao,$inserir);
12
13     mysqli_close($conexao);
14
15     ?>
16     <p align="center"><a href="aluno.php">Voltar</a></p>
```

Arquivo cadastro_aluno.php

Essa será a tela para cadastro dos alunos. Perceba que temos um *select* para selecionar o curso desejado. Os cursos que estão cadastrados aparecem todos como opção para escolha do usuário. Após o cadastramento podemos verificar e ter a certeza de que os dados foram armazenados.

Tela do site para cadastro de alunos



Tela do phpMyAdmin com os dados dos alunos cadastrados.

Apêndice 01

PHP – Conceitos Básicos

Quer criar um site dinâmico capaz de interação com o usuário através de formulários, e capaz de trabalhar também com banco de dados e demais aplicações? O PHP vai ajudar você a criar programas para a web de maneira rápida e fácil. Ela pode ser utilizada em várias plataformas de sistemas operacionais e possui uma sintaxe muito simples e de muita eficiência. Basta instalar um servidor web localmente. O passo a passo dessa instalação está mais adiante em nosso livro. O PHP não tem custos para o programador. Sua licença de uso e edição é Open Source, ou seja, o código fonte é aberto e isso é uma vantagem significativa.

O ambiente de Programação PHP

Podemos utilizar várias IDE's (Ambiente de Desenvolvimento Integrado) para digitar e construir nossos códigos em PHP. Eles são softwares criados para escrevermos os códigos que desejamos, contendo todas as ferramentas necessárias para que possamos desenvolver nossos trabalhos.

Porém, se você quiser digitar seus programas PHP em um editor simples como o bloco de notas do Windows por exemplo, fique à vontade. Também irá funcionar sem problemas.

Fica a critério do desenvolvedor essa escolha, de acordo com sua necessidade e hábito de trabalho.

Usaremos nos exemplos aqui estudados o Sublime Text. Ele é um editor de código fonte muito bem aceito no mercado de desenvolvimento de programação. Ele é rápido e leve, com muitos recursos e funcionalidades, com realce na sintaxe dos códigos e com facilidade na verificação de bugs. Você pode baixar o Sublime Text no link abaixo:

<https://www.sublimetext.com/download>

Vamos utilizar também o XAMPP que é um pacote com os principais servidores de código aberto existentes no mercado de informática. Ele será importante pois todos os arquivos gerados devem ficar na pasta htdocs: C:\xampp\htdocs.

O XAMPP pode ser baixado no endereço abaixo:

https://www.apachefriends.org/pt_br/download.html

Vamos ver a seguir as principais características da linguagem PHP.

Tipos de Dados

Dados são as informações processadas pelo computador e podem ser de vários tipos.

- **Inteiros:** usamos esse tipo que corresponde aos números que pertencem aos conjuntos \mathbb{N} e \mathbb{Z} . Não possuem casas decimais e podem ser negativos.

Ex.: 25, -57 e 0;

- **Ponto flutuante:** usamos esse tipo que corresponde aos números que pertencem aos conjuntos \mathbb{R} e \mathbb{Q} . Podem possuir casas decimais e podem ser positivos ou negativos.

Ex.: 0.01, 12.1 e -37.57;

- **Strings:** usamos esse tipo quando precisamos de dados formados por uma sequência de caracteres, que podem ser letras, números ou símbolos especiais. Também são chamados de alfanuméricos, caracteres ou literais, e podem facilmente ser representados e delimitados por aspas simples.

Ex.: 'Rua 02, nº 123';

- **Booleano:** expressam variáveis com valores VERDADEIRO ou FALSO. A Linguagem PHP utilizará True para verdadeiro e false para quando a expressão for considerada falsa.

- **Arrays:** São dados do tipo vetores. Podem armazenar uma lista de valores, sendo identificados pelo seu índice nessa lista

Variável

A quantidade de memória para armazenar uma informação varia de acordo com o tipo de dado, tipo de computador e tipo de linguagem. Variável é o nome dado a uma entidade destinada a guardar uma informação. Possui três atributos: **Nome, Tipo de Dado e Informação**. E diferentemente de outras linguagens o PHP sempre utiliza o caractere “\$” antecedendo o nome que desejamos definir para uma variável. Por exemplo, se quero definir o nome **CONTADOR** para um programa devo defini-la como **\$CONTADOR**.

- **Nome:** diferencia uma variável de outra. Cada linguagem tem sua regra de criação de variáveis. Para a linguagem PHP vamos considerar o seguinte:
 1. Uma variável pode ter um ou mais caracteres na formação de seu nome;
 2. O nome deve começar com uma letra e, de nenhuma maneira poderá se iniciar com um número. Veja isso: \$NUM10 é válido; \$10NUM é inválido;

3. Dê um nome a sua variável de forma que não tenha espaços em branco e nem utilize caracteres que não sejam letras e números. Porém, podemos utilizar o caractere underscore. Veja como podemos utilizá-lo corretamente: `$VALOR_TAXA`. Assim evitamos o espaço em branco entre `$VALOR` e `TAXA`, utilizando um caractere permitido;
4. Um outro detalhe muito importante em linguagem PHP é que ele diferencia caracteres maiúsculos dos minúsculos. Assim sendo, quando ele “olhar” para a variável com nome `$VALOR`, ele irá diferenciar da variável `$Valor` por exemplo.

O conceito de variável facilita a programação, permitindo acessar dados por meio de um nome, em vez do endereço de uma célula.

Operadores Aritméticos

São aqueles que o resultado da avaliação é do tipo numérico. Somente é permitido o uso com variáveis e constantes numéricas.

<p>+ - * /</p>	<p>Operadores aritméticos tradicionais de adição, subtração, multiplicação e divisão. Por convenção, * e / têm precedência sobre + e -. Para modificar a ordem de avaliação das operações, é necessário usar parênteses como em qualquer expressão aritmética. Vamos ver alguns exemplos de utilização desses operadores: \$b + 2; \$a - \$c; \$n * 2 e \$T / \$p.</p>
<p>%</p>	<p>Operador de módulo (isto é, resto da divisão inteira). Por exemplo, 5 % 2 = 1, ou se utilizarmos uma variável ficará dessa maneira: \$a % \$c que será o resto da divisão de \$a pôr \$c. Tem a mesma precedência do operador de divisão tradicional.</p>
<p>Alguns exemplos da saída que será exibida na utilização do operador de módulo. Em breve estudaremos os demais símbolos que estão em cores diferentes dos cálculos apresentados.</p> <pre data-bbox="130 1241 842 1538"><?php echo (7 % 3)."\n"; // imprime 1 echo (5 % 2)."\n"; // imprime 1 echo (5 % 3)."\n"; // imprime 2 echo (-5 % -3)."\n"; // imprime -2 echo (1 % 2)."\n"; // imprime 1 ?></pre>	

Operadores Relacionais

São aqueles utilizados para efetuar uma comparação. Fornece como resultado um valor lógico. São eles:

Operador	Comparação
==	Igual
!=	diferente
<	menor
>	maior
<=	menor igual a
>=	maior igual a

Operadores Lógicos

São aqueles cujo resultado da avaliação é um valor lógico.

!	Operador de negação não VERDADEIRO = FALSO, e não FALSO = VERDADEIRO. Esse operador inverte o estado lógico de uma condição
	Operador que resulta VERDADEIRO quando um dos seus relacionamentos lógicos for verdadeiro. Esse símbolo é obtido apertando o shift e a tecla de barra invertida ao lado da letra Z.
&&	Operador que resulta VERDADEIRO somente se seus dois ou mais relacionamentos lógicos forem simultaneamente verdadeiros.

Exemplo: Dadas as variáveis **A** e **B** sendo lógicas, abaixo temos todas as possibilidades de avaliação:

A	B	!A	!B	A && B	A B
V	V	F	F	V	V
V	F	F	V	F	V
F	V	V	F	F	V
F	F	V	V	F	F

A tabela anterior é conhecida como **Tabela Verdade**. Podemos concluir, que:

- O operador **!** inverte o valor do operando.
- O operador **||** tem como resultado **F** somente se todos os operandos avaliados forem também **F**.
- O operador **&&** tem como resultado **V** somente se todos os operandos avaliados forem também **V**.

Referências

ABUD Junior, Gilberto; GONÇALVES, Reginaldo Luiz. Apostila de Banco de Dados - Conceitos Básicos. Home Editora. Belém-Pará, 2023.

ABUD Junior, Gilberto; GONÇALVES, Reginaldo Luiz. Apostila de Estruturas de Repetição em C# - Versão 1.0. Home Editora. Belém-Pará, 2023.

ALVES, Willian Pereira: Construindo uma Aplicação Completa com PHP e MySQL. Editora Novatec. 2017. São Paulo.

DATE, C.J.: Introdução a Sistemas de Banco de Dados. Editora GEN LTC. 2004. São Paulo.

GOMES, Ana Laura; MARTELLI, Richard: HTML5 e CSS3. Editora SENAC. São Paulo.

GONÇALVES, Reginaldo Luiz; Abud, Gilberto J. Algoritmos e Programação: Exemplos Práticos. 1ª Edição. Volume 1. Worges Editoração. Belém-Pará. 2022.

GONÇALVES, Reginaldo Luiz; Abud, Gilberto J. Programação em Linguagem C: Exemplos Práticos. 1ª Edição. Volume 2. Worges Editoração. Belém-Pará. 2022.

GONÇALVES, Reginaldo Luiz; Abud, Gilberto J. Programando em JavaScript: Exemplos Práticos. 1ª Edição. Volume 3. Worges Editoração. Belém-Pará. 2022.

GONÇALVES, Reginaldo Luiz; Abud, Gilberto J. Programando em PHP: Exemplos Práticos. 1ª Edição. Volume 4. Worges Editoração. Belém-Pará. 2022.

LACERDA, Ivan Marx Freire; OLIVEIRA, Ana Liz Souto: Programador WEB: Um guia para programação e manipulação de banco de dados. Editora SENAC. São Paulo

MACHADO, Felipe; ABREU, Mauricio: Projeto de Banco de Dados: Uma visão Prática. Editora Érica Ltda. 2018. São Paulo.

Sites

Acesso em 16/04/2023:

https://www.apachefriends.org/pt_br/download.html.

Acesso em 16/04/2023: <https://www.sublimetext.com/>

Comentários finais

Convido a todos a lerem e estudarem também os nossos livros da **Coleção Programando: Aprenda Rápido**, Algoritmos e Programação: Exemplos Práticos, Programação em Linguagem C: Exemplos Práticos, Programando em JavaScript: Exemplos Práticos e Programando em PHP: Exemplos Práticos. Os recursos do PHP unido ao JavaScript, HTML e CSS são inúmeros e merecem ser estudados.

Leiam também as Apostilas de C#: Conceitos Básicos – Versão 1.0, Apostila de SQL: Conceitos Básicos – Versão 1.0, Apostila de Estruturas de Repetição em C# - Versão 1.0 e a Apostila de Banco de Dados – Conceitos Básicos.

Bons estudos aos leitores!

Apostila de PHP e MySQL - Versão 1.0

Esta apostila tem o objetivo de atender uma demanda com mais exemplos práticos no que diz respeito a construção de Sites com conexão por meio do PHP e MySQL, utilizando o XAMPP, que precisam ser bem entendidos. Além da explicação teórica rápida, acompanha alguns exemplos práticos desenvolvidos em sala de aula com os alunos dos Cursos Técnicos em Informática, Informática para Internet e Desenvolvimento de Sistemas. A metodologia de resolução de problemas é aplicada e cada atividade aqui resolvida foi estudada, explicada e construída pelo professor e pelos alunos dos respectivos cursos.

Home Editora
CNPJ: 39.242.488/0002-80
www.homeeditora.com
contato@homeeditora.com
9198473-5110
Av. Augusto Montenegro, 4120 - Parque Verde, Belém - PA, 66635-110

