

APLICAÇÕES WEB LINUX

O que você vai aprender:

Nesta aula de MariaDB e PHP, você aprenderá a criar bancos de dados, gerenciar tabelas, inserir e consultar dados usando o MariaDB. Em seguida, aprenderá como conectar o PHP a esse banco de dados e realizar operações CRUD (Create, Read, Update, Delete) para criar aplicações web dinâmicas e interativas

Servidor WEB

Um servidor web é um programa de computador que armazena arquivos e fornece acesso a eles através da Internet. Os servidores web são essenciais para a World Wide Web, pois permitem que os usuários acessem sites, aplicativos e outros recursos da Internet.

Alguns dos servidores web mais populares incluem:

- **Apache:** O Apache é o servidor web mais popular do mundo. É um servidor web gratuito e de código aberto que é conhecido por sua estabilidade e desempenho.
- **Nginx:** O Nginx é um servidor web moderno e eficiente que é conhecido por seu uso de recursos reduzido.
- **IIS:** O IIS é um servidor web da Microsoft que é pré-instalado em sistemas operacionais Windows.

PHP

PHP é uma linguagem de programação interpretada, livre e de código aberto, usada principalmente para o desenvolvimento de aplicações web dinâmicas. Foi criada por Rasmus Lerdorf em 1994 e desde então tornou-se uma das linguagens de programação web mais populares do mundo.

```
apt install php8.2 php8.2-mysql mariadb-server
```

```
php -v
```

Criar o arquivo info.php

```
<?php  
phpinfo();  
?>
```

Sintaxe Básica

O código PHP deve estar dentro das tags `<?php` e `?>`:

```
<?php  
echo "Olá Mundo!";  
?>
```

Comentários

// Comentário de uma linha

Comentário de múltiplas linhas

```
/*  
Comentário  
de múltiplas  
linhas  
*/
```

Banco de dados MariaDB

MariaDB é um SGBD relacional de código aberto derivado do MySQL. Isso significa que ele surgiu como uma ramificação do MySQL, criado por desenvolvedores originais do MySQL, com o objetivo de manter o projeto livre e aberto.

Ele é amplamente utilizado em aplicações web, servidores de banco de dados e outras soluções que exigem armazenamento e gerenciamento de dados estruturados .

Por que usar MariaDB?

Código Aberto: Gratuito e com uma grande comunidade ativa.

Confiabilidade e Desempenho: Robusto e com bom desempenho para diversas aplicações.

Compatibilidade: Compatível com MySQL, facilitando a migração.

Recursos: Possui uma variedade de recursos para gerenciamento de dados, segurança e replicação.

Como configurar um banco de dados MariaDB

Instalando o MariaDB no Linux

```
apt update
```

```
apt install mariadb-server
```

Após a instalação, o serviço MariaDB pode não iniciar automaticamente. Para iniciar e habilitar para iniciar automaticamente no boot:

```
systemctl start mariadb
```

```
systemctl enable mariadb
```

```
systemctl status mariadb
```

```
ss -nat
```

```
lsof -i :3306
```

Acessando o MariaDB

O MariaDB vem com um utilitário de linha de comando chamado mysql (apesar do nome, ele funciona perfeitamente com o MariaDB).

Para acessar o MariaDB como o usuário root (geralmente usado para administração), você pode usar o seguinte comando:

```
mysql -u root
```

Se você tiver uma senha definida para o usuário root do MariaDB, você precisará fornecer a senha:

```
mysql -u root -p
```

O terminal exibirá Enter password: e você deverá digitar a senha (não será exibida na tela). Após digitar a senha corretamente, você entrará no shell do MariaDB. Você verá algo como:

```
MariaDB [(none)]>
```

Comandos Básicos no MariaDB

Agora que você está dentro do shell do MariaDB, vamos aprender alguns comandos essenciais:

Listar os bancos de dados existentes:

```
SHOW DATABASES;
```

Este comando mostrará uma lista dos bancos de dados que já existem no seu servidor MariaDB. Geralmente, você verá information_schema, mysql, performance_schema, e sys (estes são bancos de dados do sistema).

Criar um novo banco de dados:

```
CREATE DATABASE meu_banco_de_dados;
```

Substitua meu_banco_de_dados pelo nome que você deseja dar ao seu banco de dados. Após executar este comando, use SHOW DATABASES; para confirmar que o banco de dados foi criado.

Selecionar um banco de dados para uso:

```
USE meu_banco_de_dados;
```

Este comando diz ao MariaDB que você quer trabalhar com o banco de dados meu_banco_de_dados. Você verá a prompt mudar para MariaDB [meu_banco_de_dados]>.

Criar uma tabela:

```
CREATE TABLE clientes (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(255) NOT NULL,  
  email VARCHAR(255),  
  data_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Este comando cria uma tabela chamada clientes com as seguintes colunas:

id: Um inteiro que se incrementa automaticamente (AUTO_INCREMENT) e é a chave primária (PRIMARY KEY) da tabela. Chaves primárias identificam unicamente cada linha na tabela.

nome: Uma string de até 255 caracteres (VARCHAR) que não pode ser vazia (NOT NULL).

email: Uma string de até 255 caracteres.

data_cadastro: A data e hora em que o cliente foi cadastrado (TIMESTAMP). Por padrão (DEFAULT), recebe a data e hora atual (CURRENT_TIMESTAMP).

Listar as tabelas no banco de dados atual:

```
SHOW TABLES;
```

Este comando mostrará uma lista das tabelas no banco de dados que você selecionou com o comando USE.

Descrever a estrutura de uma tabela:

DESCRIBE clientes;

Ou, de forma abreviada:

DESC clientes;

Este comando mostrará detalhes sobre as colunas da tabela clientes, como o tipo de dado, se é chave primária, etc.

Inserir dados em uma tabela:

```
INSERT INTO clientes (nome, email) VALUES ('João Silva', 'joao.silva@example.com');
```

```
INSERT INTO clientes (nome, email) VALUES ('Maria Souza',  
'maria.souza@example.com');
```

Este comando insere duas novas linhas na tabela clientes, com os valores especificados para as colunas nome e email. A coluna id será preenchida automaticamente com um valor único, e a coluna data_cadastro será preenchida com a data e hora atual.

Inserir vários registros de uma vez

```
INSERT INTO clientes (nome, email, telefone) VALUES ('Marta',  
'marta@email.com','2153309948'), ('Pedro', 'pedro@email.com','4453243243');
```

Selecionar dados de uma tabela:

```
SELECT * FROM clientes;
```

Este comando seleciona todas as colunas (*) de todas as linhas da tabela clientes. Você verá os dados que você inseriu.

Atualizar a tabela;

```
ALTER TABLE clientes ADD telefone VARCHAR(20);
```

```
desc clientes;
```

Selecionar dados com condição:

```
SELECT * FROM clientes WHERE nome = 'João Silva';
```

Este comando seleciona todas as colunas da linha onde o nome é 'João Silva'.

Atualizar dados em uma tabela:

```
UPDATE clientes SET email = 'joao.silva.atualizado@example.com' WHERE nome = 'João Silva';
```

Este comando atualiza o email do cliente 'João Silva'.

Deletar dados de uma tabela:

```
DELETE FROM clientes WHERE nome = 'João Silva';
```

```
DELETE FROM usuarios WHERE id = 2;
```

Este comando remove o cliente 'João Silva' da tabela. Cuidado ao usar este comando! Se você não especificar uma condição (WHERE), você apagará todas as linhas da tabela.

Apagar uma tabela:

```
DROP TABLE clientes;
```

Este comando remove a tabela clientes completamente do banco de dados.

Apagar um banco de dados:

```
DROP DATABASE meu_banco_de_dados;
```

Este comando remove o banco de dados meu_banco_de_dados e todas as tabelas contidas nele. Cuidado ao usar este comando! Não há como desfazer esta ação.

Sair do shell do MariaDB:

```
exit
```

```
quit
```

GERENCIAR DE USUÁRIOS

Segurança - Definindo uma Senha para o Usuário Root

Por padrão, o usuário root do MariaDB pode não ter senha. É altamente recomendável definir uma senha para o usuário root por motivos de segurança.

Acesse o MariaDB como root:

```
mysql -u root
```

Defina a senha de root:

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'sua_senha_segura';
```

```
FLUSH PRIVILEGES;
```

Descrição das opções deste comando anterior:

ALTER USER: Modifica as propriedades de um usuário.

'root'@'localhost': Especifica o usuário root que está acessando o MariaDB localmente.

IDENTIFIED BY 'sua_senha_segura': Define a nova senha.

FLUSH PRIVILEGES: Recarrega as tabelas de privilégios para que as alterações tenham efeito imediato.

Saia do MariaDB:

```
exit
```

Acesse novamente o MariaDB com a senha do usuário root:

```
mysql -u root -p
```

Criar novos usuários:

```
CREATE USER 'use'@'localhost' IDENTIFIED BY '123';
```

```
GRANT ALL PRIVILEGES ON *.* TO 'use'@'localhost';
```

```
FLUSH PRIVILEGES;
```

Visualizar usuários:

```
SELECT User, Host FROM mysql.user;
```

Excluir usuários:

```
DROP USER 'nome_do_usuario'@'localhost';
```

Para alterar a senha do usuário:

```
ALTER USER 'nome_do_usuario'@'host' IDENTIFIED BY 'nova_senha';
```

Verificar permissões:

```
SHOW GRANTS FOR 'nome_do_usuario'@'localhost';
```

Revogar permissões:

```
REVOKE SELECT, INSERT, UPDATE, DELETE ON database_name.* FROM  
'novo_usuario'@'localhost';
```

Descrição dos comandos gerais do Mysql:

- ALL – Permite acesso completo a um banco de dados específico. Se um banco de dados não for especificado, permite acesso completo a todo o MySQL/MariaDB.
- CREATE – Permite que um usuário crie bancos de dados e tabelas.
- DELETE – Permite que um usuário exclua linhas de uma tabela.
- DROP – Permite que um usuário exclua bancos de dados e tabelas.
- EXECUTE – Permite que um usuário execute rotinas armazenadas (stored procedures e functions).
- GRANT OPTION – Permite que um usuário conceda ou remova privilégios de outros usuários.
- INSERT – Permite que um usuário insira linhas em uma tabela.
- SELECT – Permite que um usuário selecione dados de um banco de dados.
- SHOW DATABASES – Permite que um usuário visualize uma lista de todos os bancos de dados.
- UPDATE – Permite que um usuário atualize linhas em uma tabela.

CÓDIGOS PHP

Comandos de teste na linha de comandos:

```
mysql -h localhost -u root -p meu_banco_de_dados
```

```
mysql -h localhost -u root -p baseDB
```

Código de teste de Conexão com banco:

```
<html>
<body>

<?php
// Habilita a exibição de erros
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

$servername = "localhost"; // ou o endereço do seu servidor MariaDB
$username = "root";
$password = "123"; // Substitua pela senha correta
$dbname = "base2DB";

// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifica a conexão
if ($conn->connect_error) {
    echo "Falha ao conectar ao banco de dados: " . $conn->connect_error;
} else {
    echo "Conexão com o banco de dados bem-sucedida!";
}

$conn->close();
?>

</body>
</html>
```

CÓDIGO DA APLICAÇÃO DE REGISTROS

Estrutura dos Arquivos:

1. **config.php**: Arquivo de configuração para as credenciais do banco de dados.
2. **index.php**: Página inicial com links para as outras páginas.
3. **visualizar.php**: Página para visualizar todos os registros da tabela clientes.
4. **adicionar.php**: Página com formulário para adicionar novos registros.
5. **pesquisar.php**: Página para pesquisar e encontrar registros por nome ou ID.
6. **excluir.php**: Página para excluir registros por nome ou ID.
7. **modificar.php**: Página para modificar registros existentes.
8. **style.css**: Arquivo CSS para estilização das páginas.
9. **logo.png**: Arquivo de imagem para a logo.

config.php

```
<?php
```

```
// Habilita a exibição de erros
```

```
ini_set('display_errors', 1);
```

```
ini_set('display_startup_errors', 1);
```

```
error_reporting(E_ALL);
```

```
$servername = "localhost";
```

```
$username = "root";
```

```
$password = "123"; // Substitua pela senha correta
```

```
$dbname = "baseDB";
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
if ($conn->connect_error) {
```

```
    die("Falha na conexão: " . $conn->connect_error);
```

```
}
```

```
?>
```

index.php

```
<!DOCTYPE html>
<html>
<head>
  <title>CRUD de Clientes</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="logo">
    
  </div>
  <h1>CRUD de Clientes</h1>
  <div class="menu">
    <a href="visualizar.php">Visualizar Registros</a>
    <a href="pesquisar.php">Pesquisar Registros</a>
    <a href="excluir.php">Excluir Registros</a>
    <a href="modificar.php">Modificar Registros</a>
    <a href="adicionar.php">Adicionar Registros</a>
  </div>
</body>
</html>
```

visualizar.php

```
<?php include 'config.php'; ?>
<!DOCTYPE html>
<html>
<head>
  <title>Visualizar Clientes</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="logo">
    
  </div>
  <h1>Visualizar Clientes</h1>

  <div class="menu">
    <a href="pesquisar.php">Pesquisar Registros</a>
    <a href="excluir.php">Excluir Registros</a>
    <a href="modificar.php">Modificar Registros</a>
    <a href="adicionar.php">Adicionar Registros</a>
    <a href="index.php">Voltar</a>
  </div>
</body>
</html>
```

</div>

<table>

<tr>

<th>ID</th>

<th>Nome</th>

<th>Email</th>

<th>Telefone</th>

<th>Data de Cadastro</th>

</tr>

<?php

\$sql = "SELECT * FROM clientes";

\$result = \$conn->query(\$sql);

if (\$result->num_rows > 0) {

while (\$row = \$result->fetch_assoc()) {

echo "<tr><td>" . \$row["id"] . "</td><td>" . \$row["nome"] . "</td><td>" . \$row["email"] .
"</td><td>" . \$row["telefone"] . "</td><td>" . \$row["data_cadastro"] . "</td></tr>";

}

} else {

echo "<tr><td colspan='5'>Nenhum registro encontrado</td></tr>";

}

?>

</table>

</body>

</html>

<?php \$conn->close(); ?>

adicionar.php

<?php include 'config.php'; ?>

<!DOCTYPE html>

<html>

<head>

<title>Adicionar Clientes</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="logo">

</div>

<h1>Adicionar Clientes</h1>

<div class="menu">

Visualizar Registros

Pesquisar Registros

```
<a href="excluir.php">Excluir Registros</a>

<a href="modificar.php">Modificar Registros</a>
<a href="index.php">Voltar</a>
</div>

<form method="POST">
  <input type="text" name="nome" placeholder="Nome">
  <input type="email" name="email" placeholder="Email">
  <input type="text" name="telefone" placeholder="Telefone">
  <button type="submit">Adicionar</button>
</form>
<br>
<br>
<?php
  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $telefone = $_POST['telefone'];

    $sql = "INSERT INTO clientes (nome, email, telefone) VALUES ('$nome', '$email',
'$telefone')";
    if ($conn->query($sql) === TRUE) {
      echo "<p>Cliente adicionado com sucesso</p>";
    } else {
      echo "<p>Erro ao adicionar cliente: " . $conn->error . "</p>";
    }
  }
?>
</body>
</html>
<?php $conn->close(); ?>
```

pesquisar.php

```
<?php include 'config.php'; ?>
<!DOCTYPE html>
<html>
<head>
  <title>Pesquisar Clientes</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="logo">
    
  </div>
  <h1>Pesquisar Clientes</h1>
```

```
<div class="menu">
  <a href="visualizar.php">Visualizar Registros</a>
  <a href="excluir.php">Excluir Registros</a>
  <a href="modificar.php">Modificar Registros</a>
  <a href="adicionar.php">Adicionar Registros</a>
  <a href="index.php">Voltar</a>
</div>

<form method="GET">
  <input type="text" name="termo" placeholder="Nome ou ID">
  <button type="submit">Pesquisar</button>
</form>

<table>
  <tr>
    <th>ID</th>
    <th>Nome</th>
    <th>Email</th>
    <th>Telefone</th>
    <th>Data de Cadastro</th>
  </tr>

  <?php
  if (isset($_GET['termo'])) {
    $termo = $_GET['termo'];
    $sql = "SELECT * FROM clientes WHERE nome LIKE '%$termo%' OR id = '$termo'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
      while ($row = $result->fetch_assoc()) {
        echo "<tr><td>" . $row["id"] . "</td><td>" . $row["nome"] . "</td><td>" . $row["email"] .
"</td><td>" . $row["telefone"] . "</td><td>" . $row["data_cadastro"] . "</td></tr>";
      }
    } else {
      echo "<tr><td colspan='5'>Nenhum registro encontrado</td></tr>";
    }
  }
  ?>
</table>

<br>
<br>

</body>
</html>
```

```
<?php $conn->close(); ?>
```

excluir.php

```
<?php include 'config.php'; ?>
<!DOCTYPE html>
<html>
<head>
  <title>Excluir Clientes</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="logo">
    
  </div>

  <h1>Excluir Clientes</h1>

  <div class="menu">
    <a href="visualizar.php">Visualizar Registros</a>
    <a href="pesquisar.php">Pesquisar Registros</a>
    <a href="modificar.php">Modificar Registros</a>
    <a href="adicionar.php">Adicionar Registros</a>
    <a href="index.php">Voltar</a>
  </div>

  <form method="GET">
    <input type="text" name="termo" placeholder="Nome ou ID">
    <button type="submit">Excluir</button>
  </form>

  <?php
  if (isset($_GET['termo'])) {
    $termo = $_GET['termo'];
    $sql = "DELETE FROM clientes WHERE nome LIKE '%$termo%' OR id = '$termo'";
    if ($conn->query($sql) === TRUE) {
      echo "<p>Registros excluídos com sucesso</p>";
    } else {
      echo "<p>Erro ao excluir registros: " . $conn->error . "</p>";
    }
  }
  ?>

  <div class="menu">
    <a href="index.php">Voltar</a>
  </div>
```

```
</body>
</html>
<?php $conn->close(); ?>
```

modificar.php

```
<?php
include 'config.php';

// Habilita a exibição de erros
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
?>
<!DOCTYPE html>
<html>
<head>
  <title>Modificar Clientes</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="logo">
    
  </div>
  <h1>Modificar Clientes</h1>

  <div class="menu">
    <a href="visualizar.php">Visualizar Registros</a>
    <a href="pesquisar.php">Pesquisar Registros</a>
    <a href="excluir.php">Excluir Registros</a>
    <a href="adicionar.php">Adicionar Registros</a>
    <a href="index.php">Voltar</a>
  </div>

  <form method="GET">
    <input type="text" name="id" placeholder="ID do cliente">
    <button type="submit">Buscar</button>
  </form>
  <?php
  if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $sql = "SELECT * FROM clientes WHERE id = '$id'";
    $result = $conn->query($sql);

    if ($result && $result->num_rows > 0) {
```



```
$row = $result->fetch_assoc();
```

```
?>
<form method="POST">
  <input type="hidden" name="id" value="<?php echo $row['id']; ?>">
  <input type="text" name="nome" value="<?php echo $row['nome']; ?>">
  <input type="email" name="email" value="<?php echo $row['email']; ?>">
  <input type="text" name="telefone" value="<?php echo $row['telefone']; ?>">
  <button type="submit">Salvar</button>
</form>
<?php
} else {
  echo "<p>Cliente não encontrado</p>";
}
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  $id = $_POST['id'];
  $nome = $_POST['nome'];
  $email = $_POST['email'];
  $telefone = $_POST['telefone'];

  $sql = "UPDATE clientes SET nome = '$nome', email = '$email', telefone = '$telefone'
  WHERE id = '$id'";
  if ($conn->query($sql) === TRUE) {
    echo "<p>Cliente atualizado com sucesso</p>";
  } else {
    echo "<p>Erro ao atualizar cliente: " . $conn->error . "</p>";
  }
}
?>
<div class="menu">
  <a href="index.php">Voltar</a>
</div>
</body>
</html>
<?php $conn->close(); ?>
```

style.css

```
body {
  font-family: sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}
```

```
.logo {
  text-align: left;
  padding: 10px;
}

.logo img {
  max-width: 100px;
  max-height: 100px;
}

h1 {
  text-align: center;
  color: #333;
}

table {
  width: 80%;
  margin: 20px auto;
  border-collapse: collapse;
}

table, th, td {
  border: 1px solid #ddd;
  padding: 8px;
  text-align: left;
}

th {
  background-color: #f2f2f2;
}

.menu {
  text-align: center;
  margin-top: 20px;
}

.menu a {
  margin: 0 10px;
  padding: 10px 20px;
  background-color: #007bff;
  color: white;
  text-decoration: none;
  border-radius: 5px;
}

form {
```



```
width: 50%;  
margin: 20px auto;  
}
```

```
form input, form button {  
  display: block;  
  width: 100%;  
  padding: 10px;  
  margin-bottom: 10px;  
  border: 1px solid #ddd;  
  border-radius: 5px;  
}
```

```
form button {  
  background-color: #007bff;  
  color: white;  
  cursor: pointer;  
}
```

